

abc

N° 7

COURS
D'INFORMATIQUE
PRATIQUE
ET FAMILIALE

INFORMATIQUE



Des mémoires pleines de données

Essayez le «Dragon 32»

L'informatique médicale

La fabrication des «puces»

EDITIONS
ATLAS

Sommaire

Matériel



Le Dragon 32 renferme une nouvelle puce pleine de promesses. Ce micro remporte un véritable succès depuis son apparition sur le marché, il y a un an.

130

Logiciel



Les fichiers manuels ont été remplacés par les banques de données, une méthode moderne de gestion de l'information.

124

Programmation basic



Comment une planification peut faciliter l'écriture d'un programme sans erreur.

134

Le marché



Les techniques de miniaturisation permettent de mettre des milliers de composants sur une minuscule puce de silicium.

121

L'ordinateur vient en aide au corps médical, assistant le médecin, l'administrateur et le patient handicapé.

126

Pourquoi un moniteur conçu pour l'ordinateur est-il préférable à un téléviseur ? Comment s'effectue l'affichage sur écran.

132

Mots de passe



Une découverte du XIX^e siècle est à la base de la logique informatique.

128

L'UC (unité centrale) est le cerveau de l'ordinateur.

138

Les pionniers



John von Neumann est l'architecte de l'ordinateur moderne.

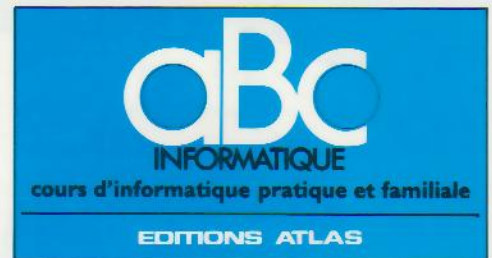
140

Prochain fascicule

- Nous analyserons le Jupiter Ace, un ordinateur peu coûteux et seul à utiliser le FORTH comme langage de programmation standard.

- Nous verrons comment les ordinateurs envahissent l'univers de la musique et ce que les ordinateurs domestiques peuvent avoir en commun avec les synthétiseurs professionnels.

- Les plans-objets permettent de créer facilement des affichages animés sur écran ; nous expliquerons comment les utiliser.



Édité par ÉDITIONS ATLAS s.a., tour Maine-Montparnasse, 33, avenue du Maine, 75755 Paris Cedex 15. Tél. : 538-52-70.

Belgique : ÉDITIONS ATLEN s.a., Bruxelles.

Canada : ÉDITIONS ATLAS CANADA Ltée, Montréal Nord.

Suisse : FINABUCH s.a., ÉDITIONS TRANSALPINES, Mezzovico.

Réalisé par EDENA s.a., 29, boulevard Edgar-Quinet, 75014 Paris. Tél. : 320-15-01.

Direction éditoriale : J.-Fr. Gautier. Service technique et artistique : F. Givone et J.-Cl. Bernar. Iconographie : J. Pierre. Correction : B. Noël.

VENTE

Les numéros parus peuvent être obtenus chez les marchands de journaux ou, à défaut, chez les éditeurs, au prix en vigueur au moment de la commande. Ils resteront en principe disponibles pendant six mois après la parution du dernier fascicule de la série. (Pour toute commande par lettre, joindre à votre courrier le règlement, majoré de 10 % de frais de port.)

Pour la France, s'adresser à ÉDITIONS ATLAS, tour Maine-Montparnasse, 33, avenue du Maine, 75755 Paris Cedex 15. Tél. : 538-52-70.

Pour les autres pays, s'adresser aux éditeurs indiqués ci-dessous.

SOUSCRIPTION

Les lecteurs désirant souscrire à l'ensemble de cet ouvrage peuvent s'adresser à :

France : DIFFUSION ATLAS, 3, rue de la Tave, 28110 Lucé. Tél. : (37) 35-40-23.

Belgique : ÉDITIONS ATLEN s.a., 55, avenue Huart-Hamoir, 1030 Bruxelles. Tél. : (02) 242-39-00. Banque Bruxelles-Lambert, compte n° 310-0018465-24 Bruxelles.

Canada : ÉDITIONS ATLAS CANADA Ltée, 11450 boulevard Albert-Hudon, Montréal Nord, H 1G 3J9.

Suisse : FINABUCH s.a., ÉDITIONS TRANSALPINES, zona industriale 6849 Mezzovico-Lugano. Tél. : (091) 95-27-44.

RELIEZ VOS FASCICULES

Des reliures mobiles, permettant de relier 12 fascicules, seront en vente en permanence chez votre marchand de journaux.

ATTENTION : ces reliures, présentées sans numérotation, sont valables indifféremment pour tous les volumes de votre collection. Vous les numéroterez vous-même à l'aide du décalque qui est fourni (avec les instructions nécessaires) dans chaque reliure.

En vente tous les vendredis. Volume I, n° 7.

ABC INFORMATIQUE est réalisé avec la collaboration de Tristan Mordrel (secrétariat de rédaction), S.I. André Laroche (traduction), Ghislaine Goullier (fabrication), Marie-Claire Jacquet (iconographie), Patrick Boman (correction). Crédit photographique, couverture : Photo SMT-Goupil.

Directeur de la publication : Paul Bernabeu. Imprimé en Belgique par Proost, à Turnhout. Distribution en France : N.M.P.P. Tax, Dépôt légal : février 1984. 17842. Dépôt légal en Belgique : D/84/2783/27.

© Orbis Publishing Ltd., London.

© Éditions Atlas, Paris, 1984.

A NOS LECTEURS

En achetant chaque semaine votre fascicule chez le même marchand de journaux, vous serez certain d'être immédiatement servi, en nous facilitant la précision de la distribution. Nous vous en remercions d'avance.

Les Éditions Atlas



Des puces et du sable

Des chercheurs ont découvert que les puces électroniques pouvaient être fabriquées à partir d'une des matières les plus abondantes.

Le silicium est plus abondant sur notre planète que tout autre élément, sauf l'oxygène avec lequel il se combine pour former du silice, matière qui nous est plus que familière. La révolution micro-électronique est faite de sable!

L'importance du silicium pour l'industrie micro-électronique vient de sa structure physique. Pur, le silicium est un conducteur médiocre, mais lorsqu'on y ajoute des quantités précises d'impuretés, il devient semi-conducteur.

La conductibilité d'une substance dépend du nombre d'électrons de ses atomes et de ses liaisons électroniques. Dans les métaux, le courant électrique est transporté par des électrons libres qui peuvent se promener à l'intérieur de la structure atomique, transférant leur charge électrique d'un atome à l'autre. Dans un corps isolant, les liaisons électroniques sont très fortes, un courant ne peut donc pas y circuler.

Le processus de fabrication du silicium pur est assez simple. D'abord, l'oxyde brut est raf-

finé jusqu'à une pureté de 99,9 %. Il est alors placé dans un creuset et chauffé jusqu'à son point de fusion, 1 410 °C, dans du gaz inerte, afin de préserver sa pureté.

L'introduction de quantités contrôlées d'impuretés (« dopage ») implique la combinaison de phosphore au silicium, ce qui produit du silicium de « type n » (parce qu'il transporte une charge négative), ou de bore, ce qui produit du silicium de « type p », ou silicium qui peut être chargé positivement.

Un cristal est produit en introduisant un cristal de semence dans le corps en fusion et en le retirant progressivement. Il est ainsi possible de produire des cristaux d'un diamètre de 7 à 10 cm et d'une longueur de 60 cm ou plus, qui sont ensuite ramenés à un diamètre standard, 76 mm ou 100 mm. Enfin, ils sont montés, coupés en tranches et passés à la meule des deux côtés, avant d'être polis d'un seul côté. La tranche terminée est généralement épaisse de 0,5 mm.



Le four à puces

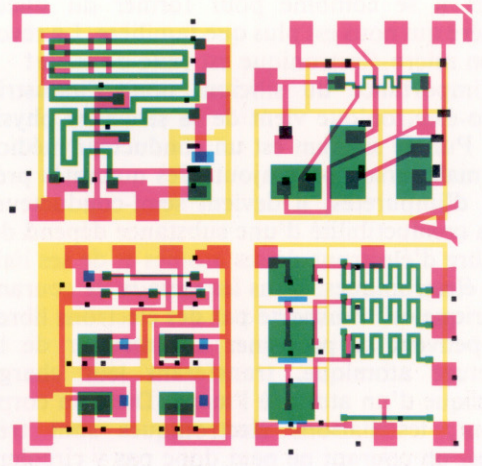
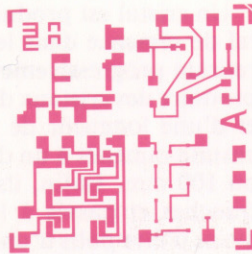
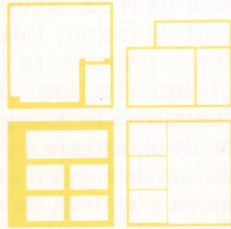
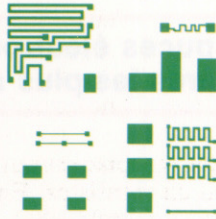
Cette photographie montre un four pour puces en fonctionnement. Après avoir été raffiné, tranché, poli, masqué et gravé, le silicium est enduit de dioxyde de silicium. Pour ce faire, il est chauffé à 1 050 °C, puis reçoit un jet d'oxygène très pur ou de vapeur surchauffée. Pendant qu'elles sont dans le four, à l'intérieur du « bateau », qui est fait de quartz en fusion, une couche de dioxyde de silicium se forme sur la surface de la tranche. Cette couche est alors enlevée sélectivement par le processus de gravure suivant, et le cycle est répété pour chaque couche de la puce. (Cl. Ferrant.)



La somme des parties

Lors de la fabrication des circuits intégrés, chaque couche du circuit doit être déposée séparément sur la tranche de silicium. Chaque photographie est d'abord séparée en trois couleurs et en noir. Ces trois couleurs sont alors soigneusement imprimées de façon cumulative pour créer la couleur désirée, selon une technique semblable à celle de l'impression couleur des journaux.

Le processus de fabrication des puces, beaucoup plus complexe que le processus d'impression, n'utilise pas d'encre mais des couches de silicium « dopé » et d'autres matériaux. Comme le montre cette illustration, les masques individuels sont combinés successivement pour construire le microcircuit, qui est ici un simple dispositif transistorisé.



Mais, le procédé étant si simple et la matière première si abondante, pourquoi la production du silicium des puces est-elle si coûteuse (150 F la tranche)?

Ce prix élevé s'explique par l'absolue nécessité de pureté du produit, qui implique des mesures très strictes. Le niveau de pureté d'air des usines de puces doit être de 3 000 particules par mètre cube — cent fois plus que dans les hôpitaux modernes.

La fabrication d'un circuit intégré implique la microgravure de la surface des puces. En production massive, cette opération est effectuée au moyen d'un procédé nommé photolithographie, semblable à celui qu'on utilise dans l'imprimerie pour produire ce périodique, par exemple.

Chaque couche du circuit est traitée séparément. Le dessin initial est réalisé à l'aide d'un ordinateur, photographié, puis réduit à la dimension réelle. Le masque est formé en reproduisant plusieurs fois cette photographie sur une matrice qui couvre toute la surface de la tranche.

Celle-là est alors chauffée à 1 050 °C dans un milieu d'oxygène pur, ce qui entraîne la formation d'une couche de dioxyde de silicium qui servira d'isolant. Puis cette couche est retirée sélectivement pour former des « fenêtres » sur le silicium pur. Le processus est répété pour chaque étape de composition du circuit intégré.

La surface oxydée est d'abord enduite d'un vernis photosensible dont la solubilité est forte-

ment réduite par exposition aux ultraviolets. Un masque de la forme de la première couche du microcircuit est introduit entre la surface enduite et la source lumineuse. La surface est exposée à une source ultraviolette, et alors « développée » dans une substance qui enlève le vernis photosensible non exposé.

Une méthode semblable est utilisée pour réaliser des gravures sur cuivre. La surface d'une feuille de cuivre est enduite de cire, sur laquelle un motif est tracé, puis la feuille est trempée dans de l'acide. Lorsqu'on enlève la cire, le motif reste gravé sur le cuivre. Aucune réaction n'a modifié le cuivre aux endroits protégés par la cire.

Après cette première étape, le processus est répété avec différents masques jusqu'à ce que le circuit soit entièrement construit. La tranche est alors enduite non sélectivement d'une autre couche de dioxyde de silicium.

Ce processus — revêtement, masquage, gravure — peut nécessiter dix opérations ou plus. Lors de chaque étape, les risques d'erreur sont élevés et le taux de rejet significatif lors de la vérification est très important. Il est proportionnel à la complexité et à la densité du microcircuit.

Lors de la première étape de vérification, la tranche, avec ses centaines de circuits intégrés identiques, doit être placée sur un appareil de contrôle commandé par ordinateur qui vérifie chaque circuit. Généralement, neuf emplacements sont réservés sur la tranche à des fins de vérification.



L'appareil de contrôle ne fait pas que marquer chaque rejet d'un point rouge; il extrait des informations concernant le taux d'erreurs au niveau de chaque tranche et des puces individuelles et note les défauts spécifiques.

Les tranches sont alors découpées en puces individuelles, et les rejets sont retirés manuellement. Les puces acceptées sont montées sur des supports métalliques miniatures.

Les broches de connexion de la puce sont reliées au support métallique par des fils très fins; l'ensemble est placé dans un boîtier en plastique ou en céramique d'où sortent les broches de connexion et subit une dernière vérification.

Actuellement, la limite du degré de miniaturisation — qui dépend de l'étape photolithographique — est égale à la longueur d'onde de la lumière, environ un demi-millionième de mètre. De récentes découvertes centrées sur l'utilisation des rayons X permettront peut-être la réduction d'un facteur 50 des circuits micro-électroniques actuels.

Les opérations de conception et de fabrication des puces sont d'une telle complexité qu'elles seraient pratiquement impossibles sans la présence de puces similaires dans les appareils qu'elles nécessitent. Quel paradoxe!

La conception assistée par ordinateur permet par exemple de prédéfinir certains sous-ensembles et de les rappeler à volonté.

Prenons l'exemple d'une puce de mémoire à accès direct (RAM), où chaque bit individuel est contenu dans une cellule de stockage à un transistor. Pour stocker 2 Koctets, la puce doit contenir un tableau de 16 384 cellules identiques. Il est assez simple pour le concepteur de définir la structure « maîtresse » et de demander à l'ordinateur de la répéter 16 384 fois.

Mais l'utilisation de l'ordinateur en conception ne s'arrête pas là. Les dessinateurs travaillent maintenant sur des terminaux et utilisent des crayons électroniques pour « dessiner » directement à l'écran. Le dessin final est préparé directement par l'ordinateur grâce à un traceur couleur.

Cette mécanisation n'améliore pas seulement le processus de conception graphique. A la page 103, nous décrivons la création du modèle informatique d'un gisement de pétrole. Une approche similaire permet au concepteur de circuit d'essayer une variété de solutions différentes avant d'entreprendre la fabrication, et il est ainsi possible d'éviter des erreurs coûteuses.

Lorsqu'on parle de révolution micro-électronique, on parle des progrès importants réalisés au niveau de la vitesse, des dimensions et du coût des composants; aussi, rappelons les caractéristiques de l'ordinateur à lampes :

- dimensions importantes;
- lenteur d'exécution;
- consommation élevée;
- mémoire et jeux d'instructions limités;
- coût élevé.

L'arrivée du transistor (voir page 46) modifia toutes ces caractéristiques, mais la fabrication

de composants était toujours très coûteuse. Ceux-ci allaient être assemblés sur des cartes de circuits imprimés. La découverte du circuit imprimé, et plus particulièrement celle du microprocesseur, permit à l'industrie de bénéficier de la fabrication commandée par ordinateur.

Certaines statistiques sont assez étonnantes. Par exemple, en 1959, un seul composant pouvait être placé sur un circuit intégré — une diode ou un transistor. En 1978, les circuits à intégration à grande échelle (LSI) les plus denses comprenaient plus d'un quart de million de composants sur une seule puce. Entre 1973 et 1983, le coût d'un bit de mémoire d'ordinateur a été divisé par vingt, et l'utilisation de composants électroniques a été multipliée par mille; ces tendances se maintiendront probablement au cours des prochaines années. On a estimé que les composants électroniques produits annuellement seront cent fois plus nombreux dans trois ans.

Actuellement, la production des puces reste l'apanage des États-Unis et du Japon. Ces deux pays ont réussi à dominer le marché, grâce à une habile politique protectionniste et à des dépenses en recherche-développement financées par



L'inventeur du microcircuit

On attribue généralement à Jack Kilby l'invention du circuit intégré, alors qu'il travaillait chez Texas Instruments en 1958. Il construisit un boîtier unique, 10 mm x 5 mm, renfermant plusieurs transistors. Les circuits électroniques modernes contiennent des centaines de milliers de dispositifs sur la même surface.

(Cl. Texas Instruments Ltée.)

l'État. Ce monopole est très dangereux pour des pays comme la France ou l'Union soviétique, qui restent très attachées à la notion d'indépendance nationale. Déjà, les États-Unis n'ont pas hésité à décréter un embargo sur certains matériels informatiques à destination de l'Union soviétique. Pour éviter tout risque futur, le gouvernement français a voulu mettre notre pays à l'abri en encourageant les industriels à produire des composants électroniques en France. Ce « Plan composants » comporte un coût financier élevé qui rend son avenir difficile à prévoir. Mais tout pays important se doit de contrôler toute la chaîne de fabrication des matériels informatiques, lesquels sont les garants de son avenir industriel et politique.



Les bases de données

Un ordinateur peut sélectionner des éléments d'information et composer des listes à partir d'une multitude d'informations stockées.

Un regroupement de données stockées et extraites par un ordinateur se nomme une « base de données ». Nous utilisons quotidiennement des bases de données non informatisées.


Un annuaire téléphonique est une base de données non informatisée. Cependant, l'information n'a pas à être stockée dans un ordre spécifique pour constituer une base de données. En fait, dans un ordinateur une telle disposition entraîne de sévères restrictions.

Un programme de base de données est un ensemble de routines qui permettent la sélection de données. Il va du système de gestion de fichiers à de véritables langages.

Une base de données renferme généralement des données nombreuses et très diversifiées. Tout ordinateur, même petit, peut gérer une base de données. L'unique restriction est la taille des dispositifs de stockage.

Par exemple, nous pouvons constituer une liste « individus » comportant des renseignements sur diverses personnes. Si nous entrons ces renseignements dans un fichier, nous obtenons une liste semblable à celle qui est nommée « fiches personnelles ». Il existe plusieurs types d'informations. Dans chaque catégorie, certains des éléments ne sont que de simples mots et certains sont des nombres, par exemple « nombre d'enfants à charge ». Il y a des restrictions dans deux catégories : « sexe » ne peut être que masculin ou féminin et « situation de famille » ne peut être que : célibataire, marié, divorcé ou veuf.



-  DISPONIBLE
-  JOUEUR DE TENNIS
-  JOUEUR DE CRICKET
-  LECTEUR
-  JOUEUR DE FOOTBALL
-  ARTISTE
-  LANCEUR DE FLÉCHETTES
-  AMATEUR DE VIN
-  PROPRIÉTAIRE DE VOITURE
-  TRANSPORTS EN COMMUN
-  CYCLISTE

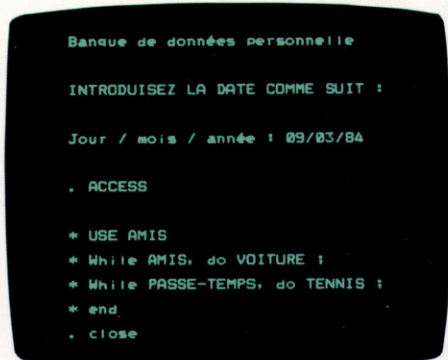
Il peut être utile de regrouper certains éléments. Par exemple : profession, nom de la société, adresse de la société, téléphone au travail et nom du supérieur pourraient être regroupés sous le titre « travail » ; la marque et l'âge de la voiture pourraient être inclus dans la liste « voiture ».

En poursuivant ce raisonnement, il est préférable de traiter les adresses en tant que listes d'éléments plutôt que de façon monolithique : nous pourrions ainsi n'extraire qu'un seul élément si nous le désirons, le principe de base étant celui de la facilité de recherche.

On cherche

un joueur de tennis

Nous avons utilisé un Rubik's Cube pour illustrer une base de données brute, c'est-à-dire une base de données sur laquelle aucune sélection ni aucun tri n'ont été faits. Dans cet exemple, nous recherchons un joueur de tennis (raquette), possédant une voiture (voiture), et qui est libre le jour en question (carrés rouges). (Cl. Tony Lodge.)

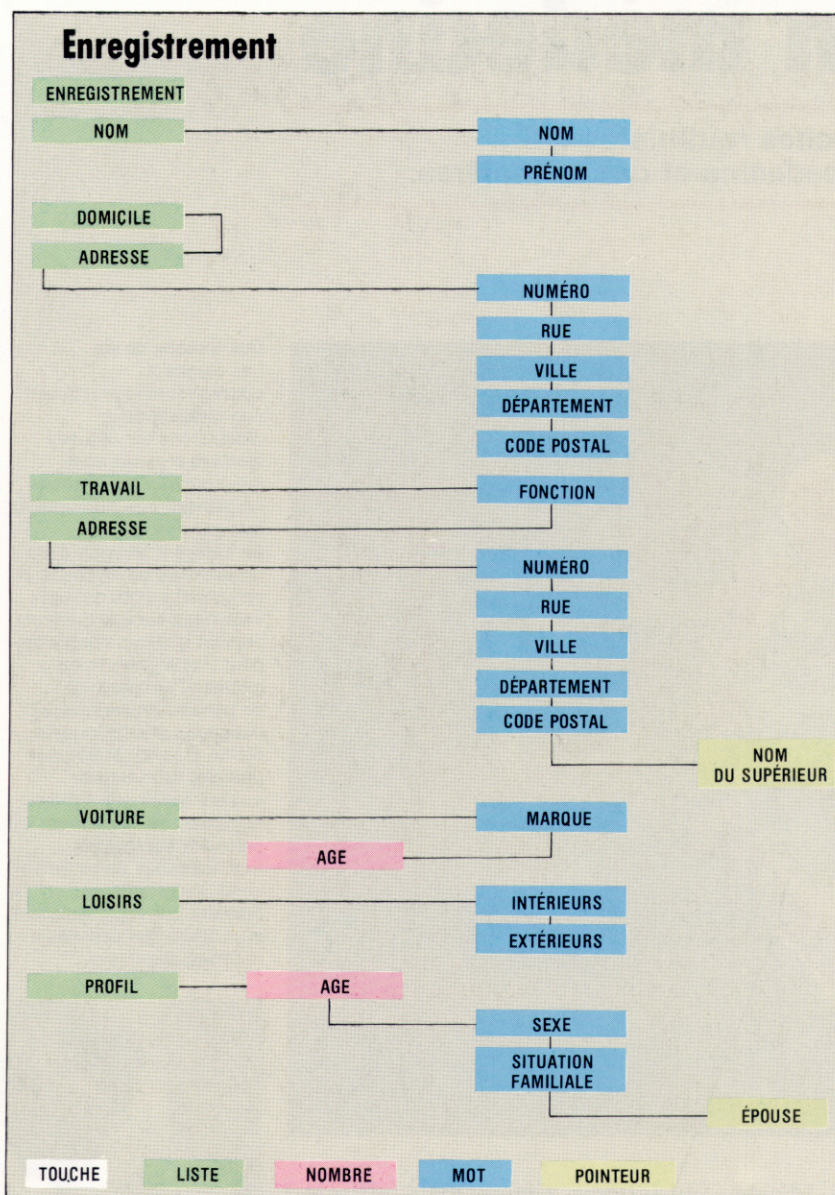


Poser les bonnes questions

La plupart des programmes de base de données utilisent des codes. ACCESS indique au programme que nous désirons « interroger » ou poser des questions au sujet d'un fichier créé antérieurement. « USE » indique que nous allons



utiliser un sous-ensemble du fichier nommé « AMIS », et nous allons extraire tous les éléments se référant à « VOITURES » ou à « TENNIS ». Nous obtiendrons une liste d'amis, avec leurs numéros de téléphone, qui jouent au tennis et possèdent une voiture. (Cl. Ian McKinnell.)



Encyclopédie moderne
Des services comme Télétel offrent l'accès à d'immenses bases de données. Il est possible, à partir d'un terminal connecté à une ligne téléphonique, d'accéder à de nombreuses applications destinées à des utilisateurs tant professionnels que privés. L'accès à une base de données est sélectionné à partir d'un menu, et l'utilisateur atteint l'information désirée en parcourant un arbre hiérarchique. Il est même possible de consulter un catalogue et d'acheter des articles par correspondance en mentionnant son numéro de carte de crédit.
(Cl. Olivier Ledoux-TDF.)



Fichier personnel

NOM
PRÉNOM
ADRESSE
TÉLÉPHONE AU DOMICILE
PROFESSION
ADRESSE AU TRAVAIL
TÉLÉPHONE AU TRAVAIL
NOM DU SUPÉRIEUR
AGE
SEXE
LOISIRS
SITUATION DE FAMILLE
TYPE DE VOITURE
AGE DE LA VOITURE

Agencement de l'information

Une base de données hiérarchisée permet à l'utilisateur de passer d'un élément d'information à un autre en pouvant choisir ces divers éléments sans avoir à connaître leur contenu.

On peut ajouter à la « situation de famille » le nom de l'épouse, lorsque c'est approprié, et n'en faire qu'un seul mot, mais puisque le fichier concerne des personnes, il est préférable d'effectuer un autre enregistrement.

Chaque enregistrement a un numéro qui correspond à sa position dans le fichier. Nous pouvons utiliser ce numéro pour désigner la personne concernée.

Cet élément de données se nomme un « pointeur ». Si nous utilisons cette même technique pour désigner le supérieur de l'individu, nous obtenons la structure « enregistrement ».

La différence entre un fichier manuel et une base de données informatisée est que, généralement, le fichier ne peut être classé qu'alphabétiquement.

Avec un fichier manuel, nous pouvons trouver le nom de la société qui emploie telle ou telle personne, mais il sera difficile de trouver les noms de tous les employés d'une société.

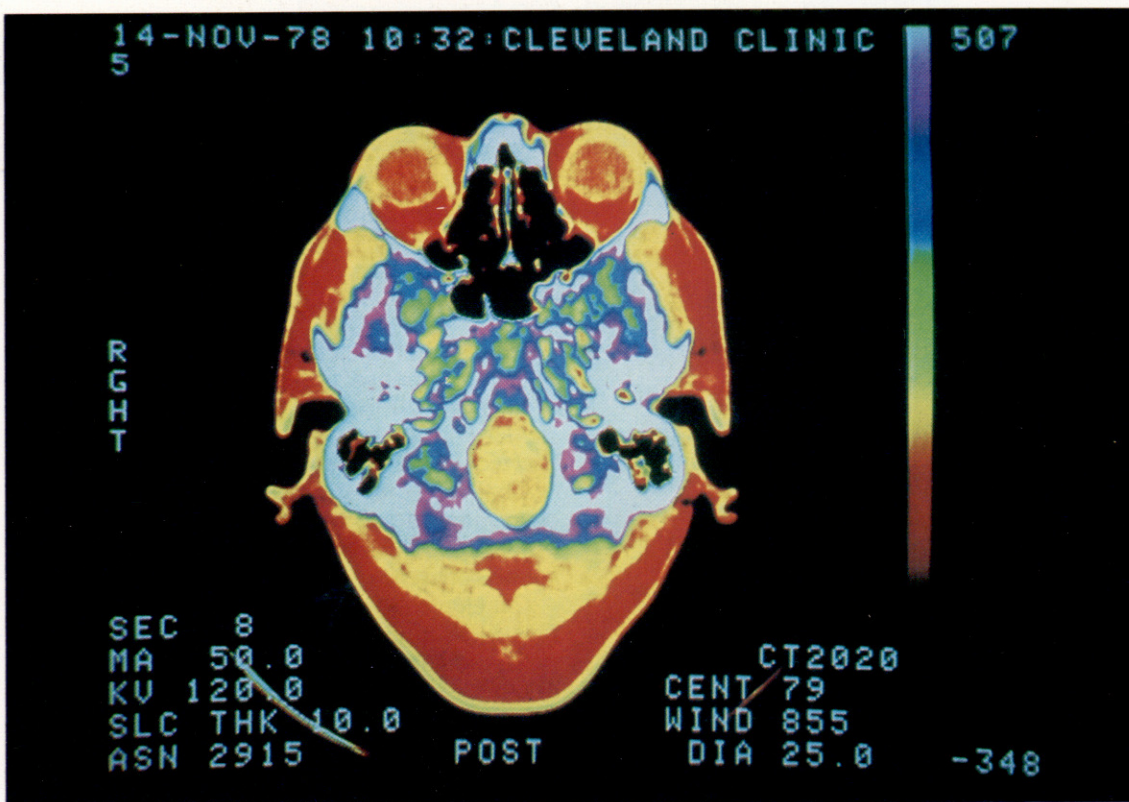
Nous pouvons consulter l'ensemble des fiches et extraire toutes celles qui sont associées aux données recherchées, mais ce travail est long et peut donner lieu à des erreurs, tandis qu'un système informatisé examine chaque enregistrement et imprime les noms des employés de la société en question.

La machine peut également définir le nom de la société comme critère de tri : la base de données contient les mêmes données, mais dans un ordre complètement différent. Ce tri regroupera tous les enregistrements concernant une société donnée. Dans un fichier manuel, il n'y a qu'un seul élément principal possible, le nom de famille. Mais dans un système informatisé, n'importe quel élément peut devenir l'élément principal.

Nous pouvons ainsi restructurer la base de données en définissant par exemple les voitures ou les noms des villes comme critère de tri. C'est le grand avantage d'un système de base de données informatisé.

Les micros en médecine

L'ordinateur a réduit la quantité de tâches routinières qui accaparaient le temps précieux des médecins et des infirmières.



Comme de nombreuses professions où un personnel hautement qualifié — et coûteux — doit consacrer une partie de son temps de travail à accomplir des tâches relativement banales, la médecine a grandement bénéficié de l'arrivée du micro-ordinateur, capable de travailler vite et bien sans risque d'erreur.

Peu après l'introduction de dispositifs équipés de microprocesseurs, les unités de soins intensifs ont considérablement modifié leurs pratiques. Les micro-ordinateurs sont rapidement devenus moniteurs de pouls, de respiration et de tension sanguine; ils permettent des lectures instantanées à partir de données multiples et libèrent ainsi le personnel infirmier qui peut de cette façon se consacrer à des tâches moins routinières.

Plus récemment, on a confié aux micros des tâches administratives, comme la mise à jour des dossiers des patients, la gestion des rendez-vous et celle des stocks pharmaceutiques.

Nous avons déjà traité une importante application médicale précédemment, mais voyons de façon plus détaillée comment des systèmes experts sont utilisés en médecine.

Mickie, un système simple mais efficace, donne au médecin un aperçu de la condition physique de son patient. Ce système n'est pas réellement expert, puisqu'il n'a pour objet que de fournir des renseignements généraux et non un diagnostic approfondi. Il ne pose au patient que des questions entraînant les réponses suivantes : « oui », « non », « je ne sais pas » ou « je ne comprends pas ». Si la réponse est « je ne comprends pas », le système reformule la question plus simplement.

Le patient n'est pas placé devant un clavier complet, il répond à l'aide d'un dispositif ne comptant que quatre touches clairement désignées. Mais ce système, dont l'exécution est plutôt lente (liée à la vitesse de lecture de chaque patient), devrait bientôt pouvoir relier les nouvelles données aux informations antérieures de façon à accélérer le diagnostic.

Les dossiers médicaux informatisés offrent un autre avantage, celui du caractère confidentiel de l'information : il est plus difficile de lire le contenu d'une disquette que de subtiliser un dossier sur papier, même si le curieux connaît l'usage de l'ordinateur.

Une tranche de vie

Les rayons X conventionnels produisent une image à deux dimensions sur laquelle tous les organes sont superposés. Ces images ne peuvent être interprétées que par des experts. En balayant le corps avec un faisceau de rayons X, et en collectant les données associées sur des capteurs puis en utilisant l'ordinateur pour traduire les signaux captés, il est possible de construire une image beaucoup plus éloquente. Même si l'ordinateur peut produire des images couleur telles que celle-ci, la plupart des radiologues préfèrent les images monochromes parce qu'ils considèrent qu'elles donnent une représentation plus claire des densités relatives des tissus.



Surveillance permanente

Le système de surveillance donne des indications sur l'état du malade, mais il peut aussi stocker des données en vue d'une analyse ultérieure. Ici, un médecin analyse des données très précises recueillies pendant le sommeil d'un patient. Ce système permet d'obtenir un meilleur diagnostic. (Cl. Ian Dobbie.)



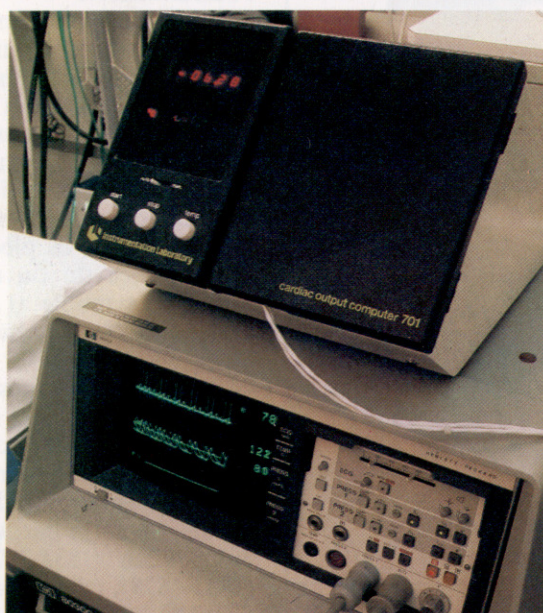
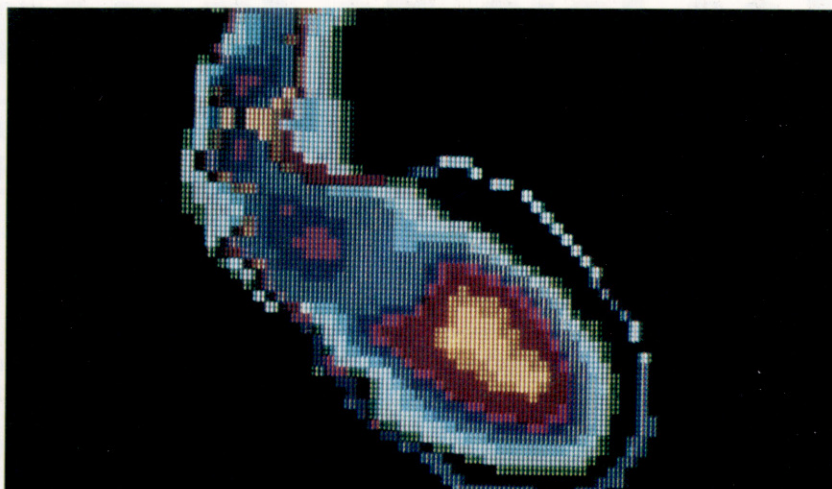
Les progrès les plus significatifs sont survenus dans le domaine du diagnostic. Jusqu'à tout récemment, toutes les méthodes d'examen interne étaient dangereuses, sinon carrément nocives : les rayons X présentaient l'éventualité d'une exposition trop intense aux radiations; l'endoscopie — insertion d'un tube de visualisation — pouvait endommager des tissus internes fragiles; et la chirurgie exploratoire comportait des risques évidents.

Les techniques tomographiques informatisées, qui utilisent un faisceau très fin de rayons X au lieu d'exposer aux radiations des parties entières du corps, représentent une amélioration importante au niveau de la précision de l'analyse et de la sécurité du malade. Les ultrasons et la résonance magnétique nucléaire laissent entrevoir la possibilité d'examens internes totalement inoffensifs.

Les systèmes d'urgence médicale bénéficient également de l'informatisation et les réussites dans le domaine des transplantations d'organes sont largement dues à l'utilisation de bases de données informatisées.

D'autres progrès spectaculaires ont été réalisés dans l'aide aux handicapés physiques et mentaux. Bien qu'encore peu développé, ce domaine offre des possibilités incroyables. Par exemple, imaginez la joie d'un muet « parlant » pour la première fois à l'aide d'un dispositif d'entrée à suivi oculaire et d'un synthétiseur de parole.

Les dispositifs de suivi oculaire mesurent les mouvements du globe oculaire. A partir d'un

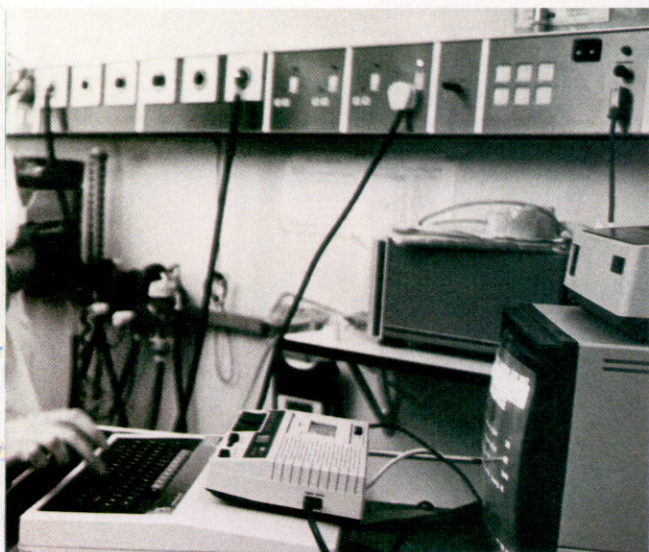


Sécurité accrue

Les analyses par résonance nucléaire magnétique, comme celle-ci, produisent des résultats semblables à ceux des analyses tomographiques informatisées, mais sont beaucoup moins dangereuses puisqu'elles n'utilisent pas de rayons X. Cette technique est toujours au stade expérimental. Une analyse peut prendre une heure.

L'état du patient sur écran

L'information sur l'état du patient, obtenue au moyen de capteurs, est interprétée par un microprocesseur, et affichée sur un oscilloscope modifié semblable à ceux qu'emploient les ingénieurs électroniciens. (Cl. Ian Dobbie.)



point de référence, il est possible de déduire quel caractère est lu.

Les puces des synthétiseurs de parole renferment en ROM les éléments de la parole, les phonèmes qui sont assemblés pour composer des mots.

Ce système peut être exploité dans le cas d'un individu totalement immobilisé. L'approche la plus répandue est de remplacer le clavier normal par une unité d'entrée personnalisée. Elle peut être, par exemple, munie de très grosses touches

afin de pouvoir être utilisée par des malades atteints de sclérose multiples ou d'ataxie locomotrice, ou encore conçue de façon à n'exiger aucune pression sur les touches.

La machine peut également proposer à l'utilisateur une série de messages affichés à un rythme spécifique, par exemple le menu de son prochain repas. En pressant un bouton au moment approprié, l'utilisateur indique à l'ordinateur ce qu'il désire manger. Ici le clavier n'est qu'un bouton pouvant être actionné par n'importe quelle partie du corps.

Cette approche pourrait même permettre à l'utilisateur de se servir d'un programme de traitement de texte et de produire, même très lentement, un travail écrit. Ne serait-ce pas là un outil précieux pour les handicapés moteurs ou pour les victimes de la thalidomide?

Il est envisagé de faire appel aux puces électroniques pour réaliser des prothèses plus performantes que celles qui existent aujourd'hui. Il serait ainsi possible de créer de vraies mains artificielles ou de permettre à des paralysés de marcher en déclenchant des stimuli nerveux commandés par un microprocesseur.

Les lois de la pensée

Un siècle avant l'invention de l'ordinateur électronique, George Boole publia des travaux concernant la logique mathématique.



George Boole est né à Lincoln, en Angleterre, en 1815. Fils de cordonnier, il devint l'un des hommes à qui nous devons l'invention de l'ordinateur. Bien qu'il soit mort en 1864, un siècle avant le début de la révolution micro-informatique, l'ordinateur moderne n'aurait probablement jamais vu le jour sans sa contribution.

Boole savait que notre processus de raisonnement quotidien peut être décrit en termes logiques. Il croyait qu'il était possible d'exprimer le raisonnement humain sous une forme mathématique. Fasciné par cette idée, il apprit les mathématiques et commença ses travaux sur la logique.

Ensembles d'informations

Supposons que vous alliez à une soirée. Vous désirez danser, cherchez une pièce où des gens dansent et essayez de trouver un partenaire.

Les personnes qui se trouvent dans la pièce dansent ou ne dansent pas. Le partenaire à qui vous demandez de danser est soit un garçon, soit une fille.

Boole aurait envisagé le problème différemment. Il aurait remarqué la présence d'« ensem-

bles » de personnes, l'ensemble masculin et l'ensemble féminin, soit M et F. Il aurait également distingué les ensembles D et V, les personnes qui dansent et celles qui veulent danser. Votre partenaire doit satisfaire à deux conditions (si vous êtes un homme) : être une fille et vouloir danser.

Boole remarqua l'importance du *et* reliant les deux conditions et lui attribua le symbole \cap . L'ensemble des partenaires possibles est représenté par l'expression $F \cap V$.

Cependant, si vous ne désirez pas danser, mais simplement bavarder avec une personne, vous pouvez la choisir parmi les ensembles M ou F qui comprennent toutes les personnes rassemblées dans la pièce. De nouveau, il nota l'importance du *ou* dans la condition et lui donna le symbole \cup . Ainsi dans son algèbre logique, $M \cup F$ représentait toutes les personnes de la pièce.

Les noms des portes logiques des ordinateurs, ET et OU, viennent des symboles de Boole. Nous découvrirons bientôt deux commandes très pratiques en BASIC, nommées AND et OR. Mais voici une représentation judicieuse de la logique booléenne, inventée par les mathématiciens anglais John Venn (1834-1923) et Charles Dodgson (1832-1898), mieux connu sous le nom de Lewis Carroll.

Examinons un problème simple. Supposons que vous ayez stocké dans la mémoire de votre ordinateur une liste de vos connaissances. Chaque nom comporte d'autres informations, numéro de téléphone, passe-temps, etc. Un après-midi, vous désirez jouer au tennis sur un court situé à l'autre bout de la ville. Vous cherchez un(e) ami(e) [pas seulement une connaissance] qui joue au tennis et possède une voiture.

George Boole (1815-1864)

Bien qu'il soit né avant que quiconque ait eu l'idée de construire un ordinateur électronique, George Boole est un des concepteurs de la logique mathématique utilisée par les ordinateurs modernes. Fils de cordonnier et autodidacte, il était convaincu que le raisonnement humain de tous les jours pouvait être exprimé sous forme de logique mathématique. Lors de la publication de ses travaux, en 1847, il devint presque immédiatement célèbre et fut invité à occuper le premier poste de professeur de mathématiques à la nouvelle université de Cork. (doc. Royal Society.)

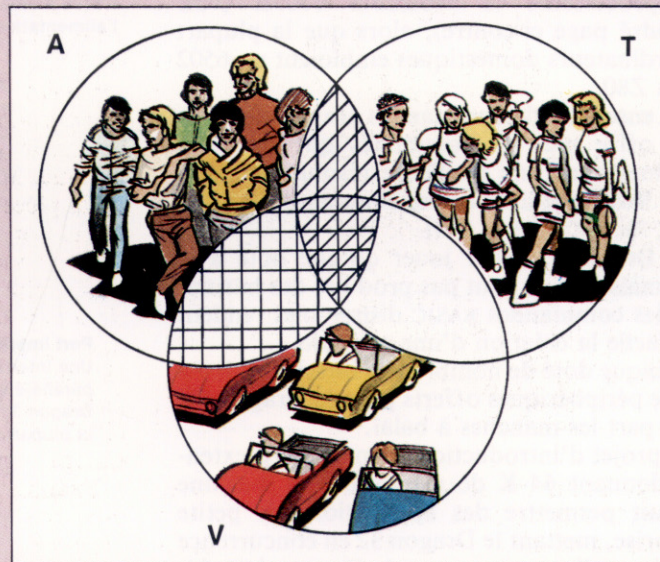
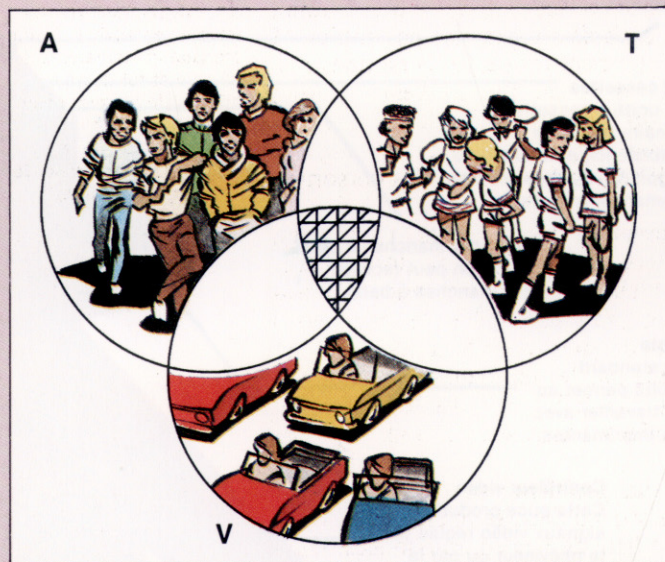
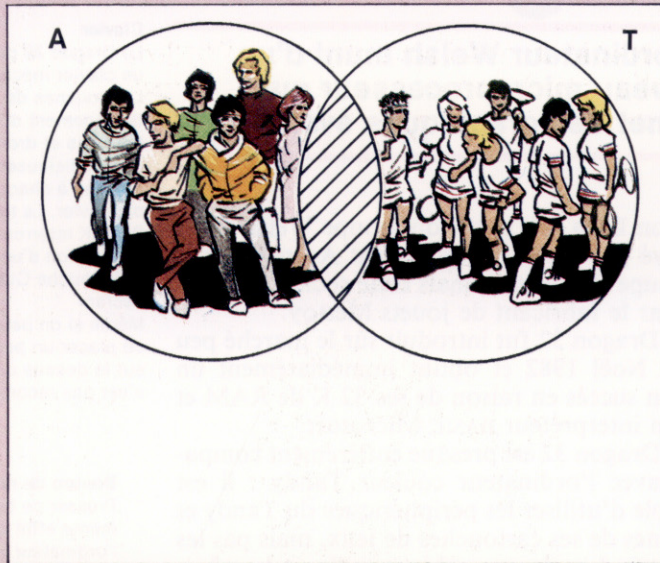
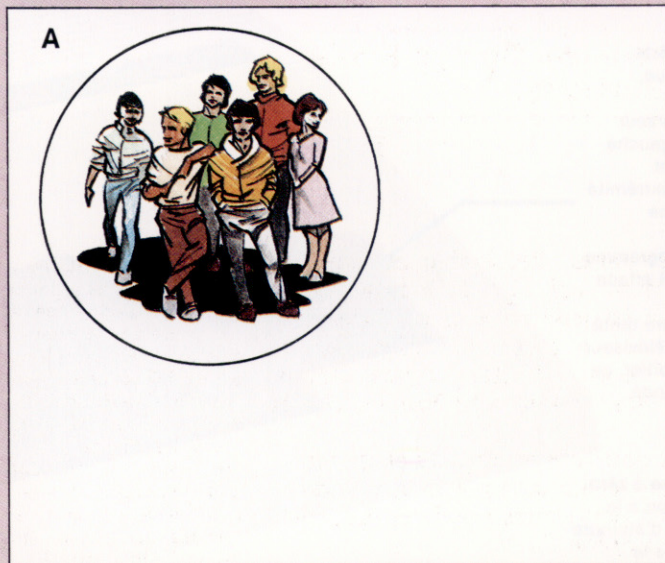
Le programme « amis »

```

10 DIM N$(10), D$(10), A$(10), T$(10), V$(10)
15 REM NOM, NO DE TEL., AMI? TENNIS? VOITURE?
17 PRINT « VEUILLEZ ENTRER LES RENSEIGNEMENTS DANS LE FORMAT SUIVANT : »
18 PRINT « NOM, NO DE TEL., OUI/NON, OUI/NON, OUI/NON »
20 FOR K = 1 TO 10
30 INPUT N$(K), D$(K), A$(K), T$(K), V$(K)
40 NEXT K
45 REM ROUTINE DE RECHERCHE
50 FOR J = 1 TO 10
60 IF A$(J) = « OUI » AND T$(J) = « OUI » AND V$(J) = « OUI » THEN GOSUB 100
70 NEXT J
80 END
100 PRINT N$(J), D$(J)
110 RETURN

```

Diagrammes de Venn



L'encadré représente l'ensemble des personnes listées dans l'ordinateur. Dans les diagrammes de Venn, c'est l'ensemble universel. Les cercles représentent les ensembles individuels. Les amis sont

placés dans le cercle A. Les joueurs de tennis sont placés dans le cercle T. Les amis qui jouent au tennis se retrouvent dans la zone de chevauchement des deux cercles ($A \cap T$). L'ensemble V représente

les propriétaires de voiture. Il chevauche les deux autres, et les personnes qui satisfont aux trois conditions se retrouvent à l'intersection des trois cercles ($A \cap T \cap V$). Le dernier diagramme

représente les mêmes ensembles, mais les conditions à remplir ont changé. Vous avez décidé que vous vouliez jouer au tennis ou aller faire une promenade en voiture. L'ensemble de vos amis

jouant au tennis est hachuré avec des lignes diagonales, tandis que l'ensemble des amis qui possèdent une voiture est hachuré avec des lignes horizontales. (Cl. Kevin Jones.)

Vous demandez à votre ordinateur d'imprimer les noms des personnes qui remplissent les trois conditions : jouer au tennis ET posséder une voiture ET être un(e) ami(e). Le programme demande d'abord des renseignements concernant chaque connaissance : est-ce un ami, possède-t-il une voiture, joue-t-il au tennis? On suppose ici que vous avez dix connaissances, mais vous pouvez changer ce nombre à volonté avec l'instruction DIM de la ligne 10. La liste est alors parcourue avec une instruction IF-THEN, dans laquelle on a introduit une condition multiple. La plupart des BASIC permettent d'insérer des conditions multiples jointes par les com-

mandes AND et OR. Finalement, les noms et numéros de téléphone des personnes qui satisfont aux trois conditions sont imprimés.

Des combinaisons très complexes de fonctions logiques sont employées dans certains programmes.

L'algèbre de Boole, que ses contemporains accueillirent comme une simple curiosité, est essentielle en informatique.

Les mathématiques modernes, si décriées dans le passé, trouvent ici une application pratique inattendue. Si les jeunes élèves « ne savent plus calculer », ils peuvent en revanche comprendre la logique de l'informatique.



Dragon 32

Un ordinateur Welsh muni d'un nouveau microprocesseur qui permet des graphiques évolués.

Dragon Data fait partie du groupe Welsh et est appuyé par Welsh Development Agency et par le groupe Prudential, mais cette société fut fondée par le fabricant de jouets Mettoy.

Le Dragon 32 fut introduit sur le marché peu avant Noël 1982 et obtint immédiatement un certain succès en raison de ses 32 K de RAM et de son interpréteur BASIC Microsoft.

Le Dragon 32 est presque entièrement compatible avec l'ordinateur couleur Tandy : il est possible d'utiliser les périphériques du Tandy et certaines de ses cartouches de jeux, mais pas les cassettes. Les deux machines utilisent le même microprocesseur, le Motorola 6809E (voir l'encadré page ci-contre), alors que la plupart des ordinateurs domestiques emploient un 6502 ou un Z80.

Un ensemble de commandes graphiques évoluées offre plus de possibilités que bien des machines avec une meilleure résolution maximale. Voici certaines de ces commandes : DRAW, CIRCLE, PAINT, COLOUR et MOVE.

Le Dragon ne peut jouer qu'une note à la fois; mais, s'il ne peut pas produire des harmonies, les commandes BASIC disponibles rendent plus facile la création d'une mélodie.

Quoique doté de nombreuses interfaces, il y a peu de périphériques offerts pour le Dragon 32, mis à part les manches à balai.

Le projet d'introduction d'une carte d'extension donnant 64 K de mémoire à la machine pourrait permettre des applications de petite entreprise, mettant le Dragon 32 en concurrence avec des ordinateurs comme le Commodore 64.

Clavier

Le Dragon 32 possède un clavier mécanique. Les touches de déplacement du curseur haut/bas et droite/gauche sont judicieusement placées à chaque extrémité du clavier. La touche BREAK interrompt l'exécution d'un programme et la touche CLEAR efface l'écran. Même si on peut être tenté de placer un petit téléviseur sur le dessus du boîtier, ce n'est pas recommandé.

Bouton de remise à zéro

Presser ce bouton a le même effet que d'éteindre l'ordinateur et de le rallumer aussitôt, mais risque moins d'abîmer l'alimentation.

Port cassettes

Une unité à cassette ordinaire peut être branchée ici, et le BASIC du Dragon 32 vous permet d'en commander le moteur.

Ports pour manches à balai

Le Dragon peut recevoir deux manches à balai.

Port imprimante

Une interface standard parallèle à 8 bits permet au Dragon 32 de travailler avec la plupart des imprimantes.

Contrôleur vidéo

Cette puce produit les signaux vidéo requis par le téléviseur ou par le moniteur à partir des caractères et symboles stockés en mémoire.

Modulateur RF

Il convertit les signaux vidéo sous une forme compatible avec l'entrée dans la prise d'antenne d'un téléviseur.

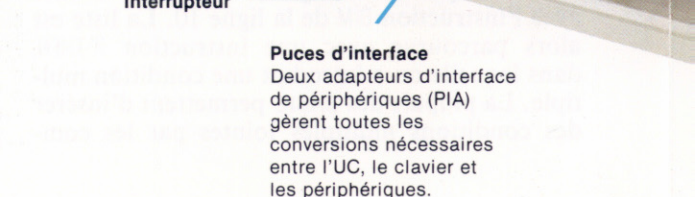
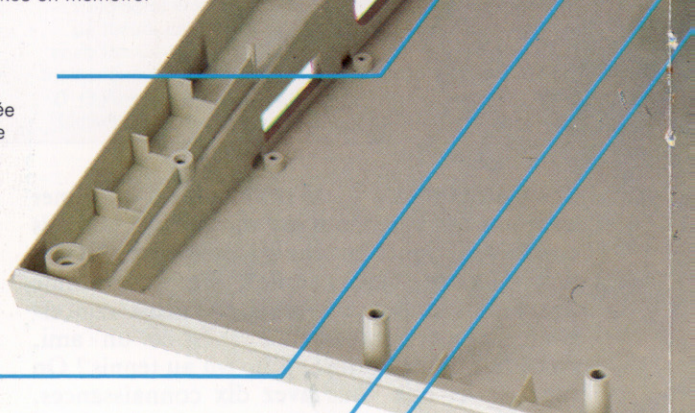
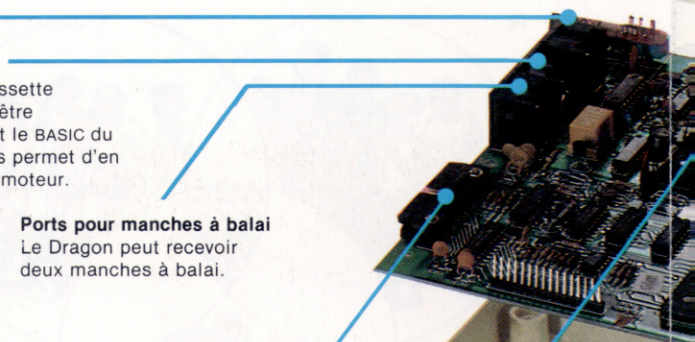
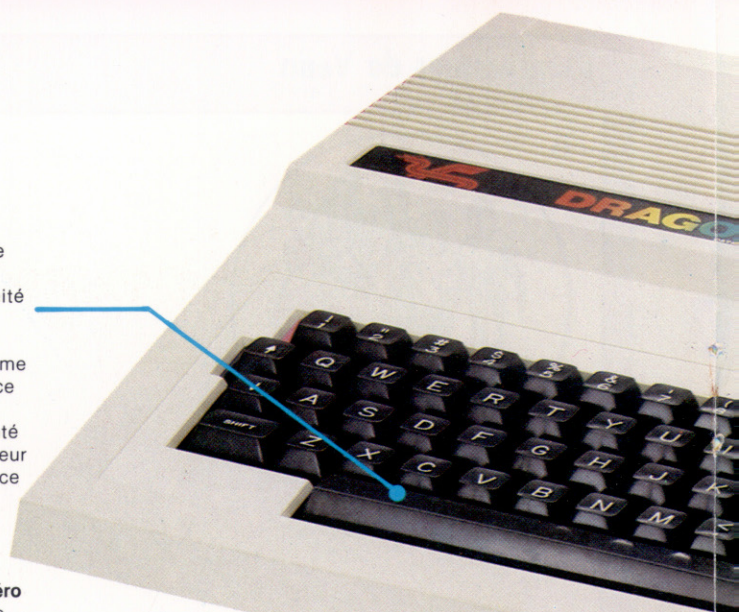
Interface vidéo

Elle sépare les signaux rouges, verts et bleus qui sont transmis ici pour commander le moniteur. Ce système est très efficace.

Interrupteur

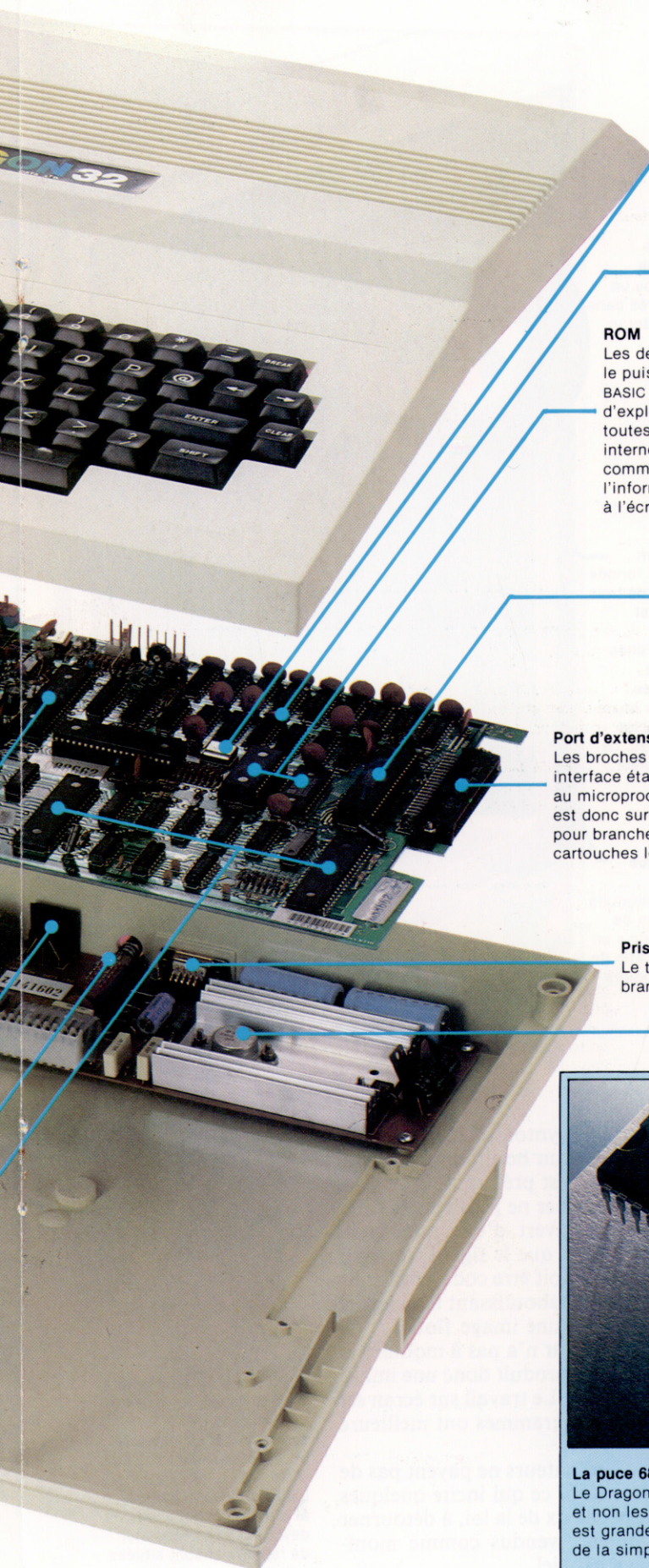
Puces d'interface

Deux adaptateurs d'interface de périphériques (PIA) gèrent toutes les conversions nécessaires entre l'UC, le clavier et les périphériques.



Lecteur de disquettes du Dragon 32

Ce lecteur est livré en deux unités : un contrôleur de disques qui se branche sur le connecteur de cartouche du Dragon, et le boîtier principal. (Clichés Chris Stevens.)

**Quartz**

Ces quartz se chargent de la synchronisation de tous les signaux. L'un commande la vitesse du microprocesseur, l'autre le circuit vidéo principalement.

RAM

La RAM standard de 32 K du Dragon 32 est contenue dans 16 puces de 2 K.

ROM

Les deux ROM contiennent le puissant interpréteur BASIC Microsoft, le système d'exploitation qui gère toutes les fonctions internes de l'ordinateur, comme le transfert de l'information du clavier à l'écran.

Microprocesseur

Le 6809 de Motorola est plus rapide que le populaire 6502, mais moins de logiciels ont été écrits pour lui. La même puce est utilisée dans l'ordinateur couleur Tandy, ce qui explique la compatibilité des deux machines.

Port d'extension principal

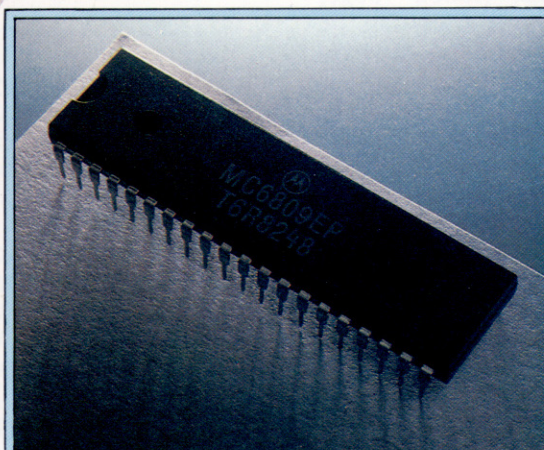
Les broches de cette interface étant connectées au microprocesseur, elle est donc surtout utilisée pour brancher des cartouches logiciel.

Régulateurs de tension

Un transformateur externe réduit la tension secteur à celle qui est utilisée par l'ordinateur. Ces régulateurs avec condensateurs cylindriques servent à aplanir les variations ou fluctuations de courant. Ils sont montés sur un dissipateur thermique métallique afin d'éviter un échauffement.

Prise d'alimentation

Le transformateur se branche à cet endroit.

**La puce 6809**

Le Dragon 32 utilise le microprocesseur à 8 bits 6809 et non les puces plus populaires 6502 et Z80. La 6809 est grandement appréciée des programmeurs en raison de la simplicité de son langage machine.

DRAGON 32**PRIX**

DIMENSIONS

380 × 325 × 97 mm

POIDS

2,100 kg

HORLOGE

1 MHz

MÉMOIRE

La RAM actuelle de 32 K sera peut-être portée à 64 K (dont seulement 48 K utilisables par le BASIC). Une ROM de 16 K renferme le BASIC de Microsoft et le système d'exploitation.

AFFICHAGE VIDÉO

16 lignes de 32 colonnes. Huit couleurs sont disponibles (uniquement deux lorsque l'ordinateur travaille en résolution graphique maximale de 256 × 192 points).

INTERFACES

Unité à cassette, manches à balai (deux ports), téléviseur, moniteur, connecteur de cartouches, interface parallèle d'imprimante, etc.

LANGAGE INTÉGRÉ

BASIC.

AUTRE LANGAGE OFFERT

Aucun.

ACCESSOIRES FOURNIS

Alimentation, prise d'antenne, manuel d'instruction.

CLAVIER

Mécanique, 53 touches.

DOCUMENTATION

Le manuel de l'utilisateur écrit par le concepteur de l'ordinateur, Richard Wadman, est excellent. Les messages affichés sont imprimés en différentes couleurs; les astuces sont nombreuses et les conseils très pratiques.

Le BASIC du Dragon 32 est expliqué, des fonctions élémentaires aux fonctions graphiques et sonores évoluées. Viennent ensuite une série d'annexes comportant entre autres des grilles d'affichage qui servent à créer des motifs graphiques.

Moniteurs

Un moniteur dédié vous permet d'obtenir une excellente qualité d'affichage pour les graphiques et les jeux.

Puisque de plus en plus de gens utilisent des ordinateurs, des vidéoscopes, et d'autres équipements nécessitant des écrans, les prix des moniteurs dédiés commencent à baisser.

Auparavant, un bon moniteur couleur pouvait coûter 9 000 ou 10 000 F, mais aujourd'hui, on peut l'acheter pour environ 5 000 F, ou même moins. Les moniteurs monochromes sont encore meilleur marché, de 1 500 à 2 200 F.

Aujourd'hui, la plupart des ordinateurs permettent d'obtenir la couleur; il est donc recommandé de se procurer un moniteur couleur.

Il existe deux types principaux de moniteurs couleur, les moniteurs RVB et les moniteurs vidéo composite. Un moniteur RVB est commandé directement, les trois canons produisant les couleurs fabriquées par l'ordinateur. Les impulsions qui synchronisent l'ordinateur et le moniteur sont aussi produites directement par l'ordinateur.

Il existe deux types d'impulsions de synchronisation, l'un pour la ligne de l'image et l'autre pour chaque image complète. A la fin de chaque balayage, le moniteur reçoit une courte impulsion l'avertissant de la fin du balayage et lui demandant de renvoyer le faisceau électronique dans le coin supérieur gauche de l'écran.

Un processus similaire se déclenche à la fin de chaque ligne, indiquant la fin de cette ligne particulière et que le faisceau électronique doit revenir du côté gauche de l'écran. Dans un moniteur RVB, ces signaux (canon rouge, canon vert, canon bleu, synchronisation de ligne, synchronisation de trame) sont envoyés au moniteur par des fils individuels.

Par contre, un moniteur composite se rapproche d'un téléviseur: tous les signaux sont combinés dans un seul puis envoyés au moniteur *via* un câble coaxial. A l'intérieur du moniteur, les signaux de synchronisation et de couleur sont séparés de nouveau pour composer l'image.

Un moniteur est un téléviseur sans syntonisateur. Il est en fait possible de convertir un moniteur en téléviseur en ajoutant un syntonisateur, ou de modifier un téléviseur ordinaire en retirant le mécanisme de syntonisation.

Cependant, cela n'est pas recommandé en raison des hautes tensions dangereuses qu'on trouve à l'intérieur de tout équipement renfermant un tube à rayon cathodique. Même un technicien professionnel ne serait pas enchanté d'avoir à résoudre ce problème. Par ailleurs, la conception des circuits est légèrement diffé-

Grille de l'écran

Afin de garantir une définition précise de l'image, une grille ou un masque sont intégrés dans la surface du tube écran.

Phosphore de l'écran

L'image couleur est formée par la combinaison de trois couleurs. Le verre est enduit de diverses substances. Lorsqu'elles sont irradiées par le faisceau électronique, elles s'illuminent en rouge, en vert ou en bleu selon l'intensité du faisceau, ce qui crée l'image couleur.

Faisceau électronique

Trois faisceaux électroniques sont produits dans le tube; chacun de ces faisceaux « excite » un phosphore différent afin de produire les divers points colorés.

rente; ainsi, même si le syntonisateur est retiré, le téléviseur ne sera pas un bon moniteur.

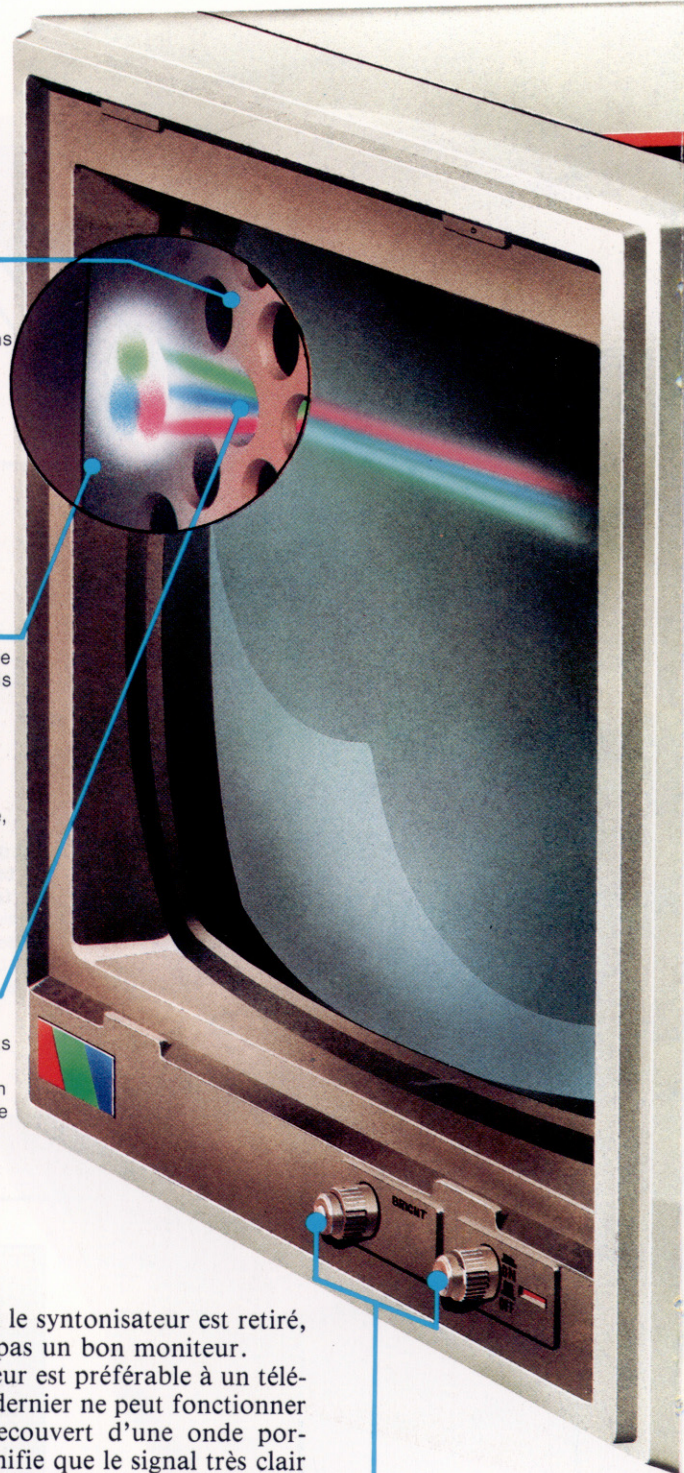
Un moniteur couleur est préférable à un téléviseur, parce que ce dernier ne peut fonctionner qu'avec un signal recouvert d'une onde porteuse UHF. Cela signifie que le signal très clair généré par l'ordinateur doit être codé, transmis, puis décodé de nouveau, aboutissant à un signal « brouillé » qui produit une image floue.

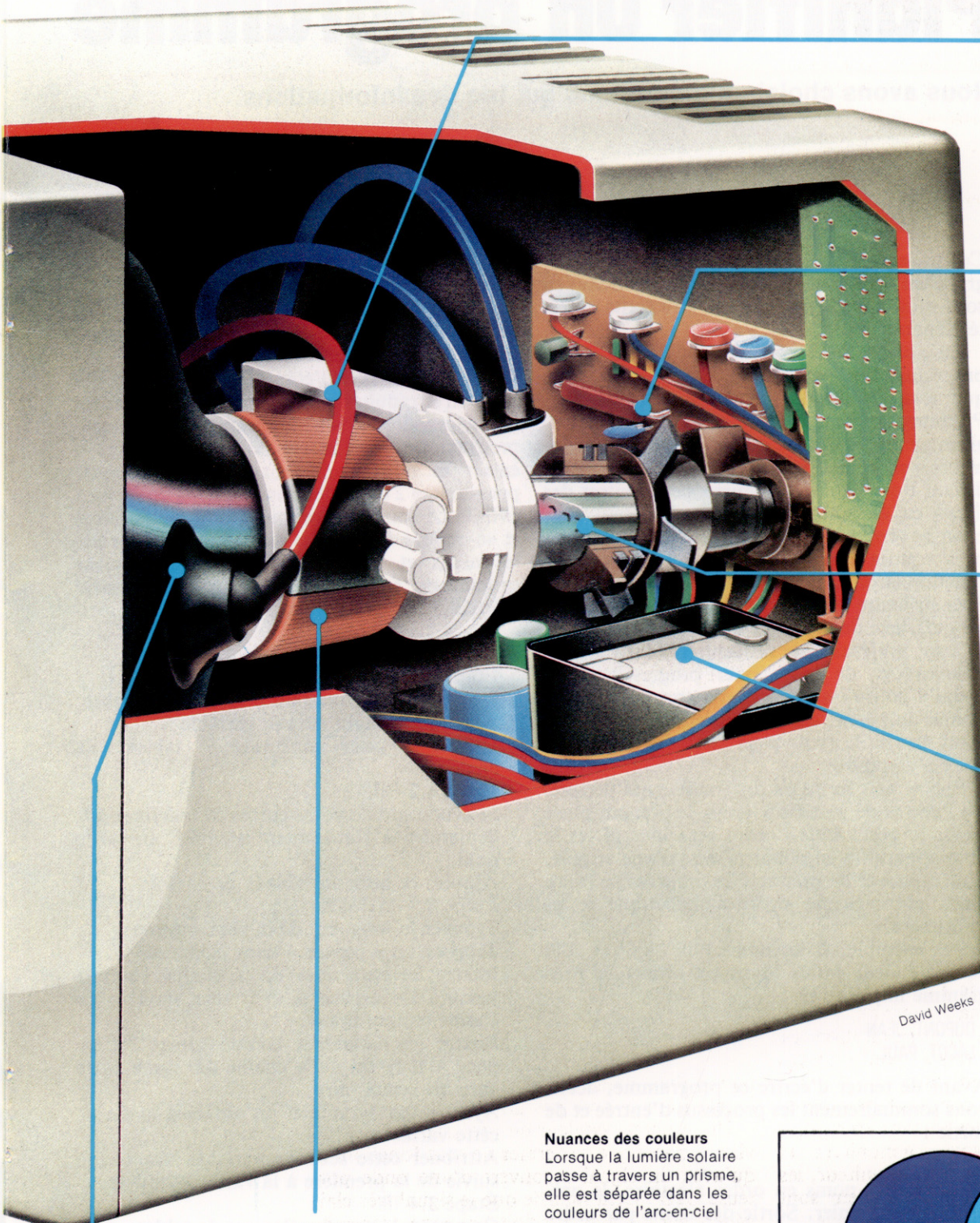
Par contre, un moniteur n'a pas à moduler et à démoduler le signal, il produit donc une image plus nette et plus précise. Le travail sur écran est moins fatigant et les programmes ont meilleure apparence.

Les moniteurs d'ordinateurs ne payent pas de redevance à la télévision, ce qui incite quelques acheteurs, peu respectueux de la loi, à détourner des téléviseurs couleur, vendus comme moniteurs, de leur usage officiel.

Boutons

Comme sur un téléviseur, il y a diverses commandes. Les boutons de synchronisation verticale et horizontale peuvent être réglés par l'utilisateur. L'intensité de la couleur et d'autres commandes ne devraient pas nécessiter de réglage et sont situées à l'intérieur du moniteur.





Circuits haute tension
Puisque les tubes à rayon cathodique nécessitent des tensions très élevées, ils doivent inclure un circuit de redressement pour élever la tension secteur (240 V) au niveau requis.

Carte du circuit principal
Les circuits commandant les mouvements du faisceau et la sollicitation des canons électroniques se trouvent ici. Une partie de la section de synchronisation de ligne, qui opère à des fréquences très élevées, peut être utilisée comme alimentation commutatrice pour le tube

Canons
Un moniteur couleur, tout comme un téléviseur couleur, a trois canons électroniques, un pour le rouge, un pour le vert et un dernier pour le bleu. Ils sont situés à l'arrière du tube.

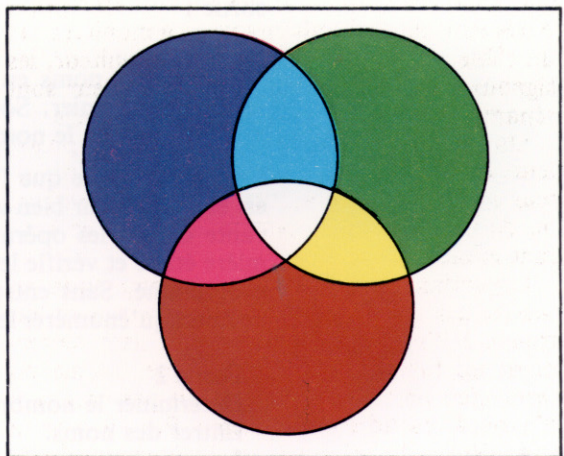
Alimentation
Un tube à rayon cathodique doit être piloté par des tensions en courant continu très stables et requiert des courants assez intenses ; un transformateur puissant est donc nécessaire.

David Weeks

Branchement de l'anode
Lorsqu'un canon émet un faisceau, celui-ci est accéléré par un champ à haute tension. Ce champ doit se trouver à l'autre extrémité du tube et est appliqué au moyen de cette plaque très bien isolée à l'extrémité du câble.

Collier
Le collier est composé de plusieurs bobines qui produisent de puissants champs magnétiques. Ceux-là varient très rapidement, de façon à déplacer le faisceau sur le phosphore, ce qui produit l'image.

Nuances des couleurs
Lorsque la lumière solaire passe à travers un prisme, elle est séparée dans les couleurs de l'arc-en-ciel (spectre) qui vont du rouge au bleu en passant par le vert. Si ce spectre traverse un autre prisme, semblable au premier, les couleurs sont recombinaées et restituent la lumière solaire initiale (souvent nommée « lumière blanche »). Ce processus de combinaison ou d'addition est utilisé dans un moniteur couleur. (Cl. Mark Watkinson.)



Planifier un programme

Nous avons choisi un programme qui trie des informations.

Ce programme illustre comment des programmes relativement complexes sont divisés en sous-programmes qui peuvent être écrits et testés séparément.

En plus des tests séparés, les sous-programmes permettent une progression logique lors du développement du programme. Plusieurs approches permettent d'écrire un programme en BASIC. La méthode « par tâtonnements » est la plus commune : vous vous asseyez devant l'ordinateur et commencez à entrer des lignes en BASIC sans avoir planifié soigneusement le programme. Cette méthode donne des programmes mal structurés qui souvent ne fonctionneront pas lors de la première tentative. Si la structure du programme n'est pas claire, il sera difficile de le mettre au point.

Il est préférable de réfléchir d'abord à cette structure et de la raffiner progressivement, jusqu'à ce qu'un programme correct et fonctionnel soit écrit. Un organigramme peut également être utile (voir page 104). Voyons comment s'y prendre.

Problème : écrire un programme qui permettra l'entrée de plusieurs noms, le prénom placé avant le nom. Puis l'ordre sera inversé, et le nom apparaîtra en premier, suivi d'une virgule, d'un espace et du prénom. Le programme triera alors les noms par ordre alphabétique et les imprimera.

Par exemple, si les noms PAUL LACOT et JEAN DUPONT étaient entrés (dans cet ordre), le programme imprimerait :

```
DUPONT, JEAN
LACOT, PAUL
```

Avant de tenter d'écrire ce programme, décrivons sommairement les processus d'entrée et de sortie :

Étape 1

Entrée des noms en ordre aléatoire, le prénom en premier. Sortie des noms par ordre alphabétique, le nom en premier.

Cela ne présente que l'objet du programme et est essentiel pour bien le structurer. L'étape suivante définit les opérations composant la première étape et vérifie le bon fonctionnement du programme. Sans entrer dans les détails, vous ne faites qu'énumérer les opérations impliquées :

Étape 2

Déterminer le nombre de noms à entrer.
Entrer des noms.
Inverser les noms.

Trier les noms.

Imprimer les noms.

Examinez cette liste. Fonctionne-t-elle correctement? Le déroulement est-il logique?

Les opérations de l'étape 2 sont assez simples pour être traduites en sous-programmes. Nommons ces divers sous-programmes. Le sous-programme 1 définit le nombre de noms à entrer et nous l'appelons NOMBRE. Le sous-programme 2 entre des noms et nous l'appelons ENTREE. Le sous-programme 3 inverse les noms et se nomme INVERSE. Le sous-programme 4 trie les noms et est appelé TRI. Enfin, le sous-programme 5 imprime les noms et est appelé IMPRESSION.

Étape 3.1 NOMBRE

Demander à l'utilisateur d'entrer un nombre.

Attribuer ce nombre à la variable N.

Utiliser N pour constituer le tableau de chaîne.

Étape 3.2 ENTREE

Si le nombre de noms entrés est inférieur à N, demander à l'utilisateur d'entrer un autre nom.

Ajouter ce nom au tableau de chaîne.

Étape 3.3 INVERSION

Trouver la longueur de la chaîne (nom).

Trouver un « espace » dans la chaîne.

Mettre les caractères de la chaîne compris jusqu'à l'« espace » dans une variable de chaîne temporaire.

Mettre les caractères compris entre l'« espace » et la fin de la chaîne dans une autre variable temporaire.

Ajouter une virgule et un espace à la fin de cette variable.

Attribuer cette seconde variable temporaire suivie de la première à la variable initiale.

Étape 3.4 TRI

Comparer le premier élément du tableau avec le suivant.

Si le premier élément est supérieur au suivant (à un niveau plus élevé dans l'ordre alphabétique), permuter.

Comparer le deuxième et le troisième élément.

Permuter si nécessaire.

Répéter jusqu'à comparaison de toutes les paires.

Revenir au début du tableau et répéter la comparaison des paires jusqu'à ce qu'aucune permutation n'ait lieu.

Remarque : cette routine de tri est exactement la même que celle qui est utilisée dans le cours de programmation précédent. La partie « permutation » sera traitée comme un sous-programme appelé à partir du sous-programme TRI.

Étape 3.5 IMPRESSION

Imprimer tous les éléments du tableau.

Nous avons maintenant élaboré de façon assez détaillée toutes les composantes de notre programme. Le sous-programme TRI n'a été que brièvement esquissé puisque nous l'avons traité lors du dernier cours. Le sous-programme PERMUTATION n'a pas été décrit. Voyons maintenant comment traduire cette première ébauche en un programme BASIC.

Étape 4

1. NOMBRE

Les trois lignes de l'étape 3.1 se traduisent directement en BASIC. Le message à l'utilisateur est produit par une instruction PRINT, le nombre est entré par une instruction INPUT et le tableau est constitué à l'aide d'une instruction DIM :

```
PRINT « COMBIEN DE NOMS DESIREZ-VOUS ENTRER ? »
INPUT N
DIM A$(N)
RETURN
```

La variable N renferme maintenant le nombre maximal de noms à entrer. L'instruction DIM déclare un tableau de chaîne. Les variables de chaîne renferment des chaînes de caractères alphanumériques et non des nombres. Le nom d'une variable de chaîne se termine toujours par un signe « dollar ». A\$(N) seul ne pourrait contenir qu'une seule chaîne, A\$(N) crée un tableau qui peut contenir N chaînes. Nous avons traité précédemment des variables indicées.

L'instruction RETURN redonne la main au programme principal à la ligne suivant l'appel du sous-programme. Les valeurs attribuées aux variables dans le sous-programme seront reportées dans le programme principal et peuvent être utilisées ailleurs dans le programme, même dans les sous-programmes.

2. ENTREE

Tant que le nombre de noms entrés est inférieur à N, un message sollicite d'autres entrées auprès de l'utilisateur et celles-ci sont ajoutées au tableau des noms. Une boucle FOR-NEXT est donc requise; le premier nom entré dans le tableau sera son premier élément, et le dernier nom sera son N^e, ainsi :

```
FOR X = 1 TO N
PRINT « ENTREZ UN NOM »
INPUT A$(X)
NEXT X
RETURN
```

Cela entrera tous les noms dans le tableau. Mais les lecteurs attentifs se demandent peut-être comment nous allons inverser l'ordre des noms et des prénoms dans le sous-programme INVERSE. Chaque élément (nom) du tableau devra être

extrait, inversé, puis remis dans le tableau. Au lieu de cette approche longue et compliquée, il serait plus simple d'appeler le sous-programme INVERSE à l'intérieur du sous-programme ENTREE après l'entrée de chaque nom. Le nom peut ainsi être inversé immédiatement et attribué au tableau. Pour ce faire, nous n'avons qu'à ajouter une ligne :

```
FOR X = 1 TO N
PRINT « ENTREZ UN NOM »
INPUT A$(X)
GOSUB (INVERSE)
NEXT X
RETURN
```

Tous les noms dans le tableau seront donc maintenant inversés (nom suivi du prénom) et prêts à être triés.

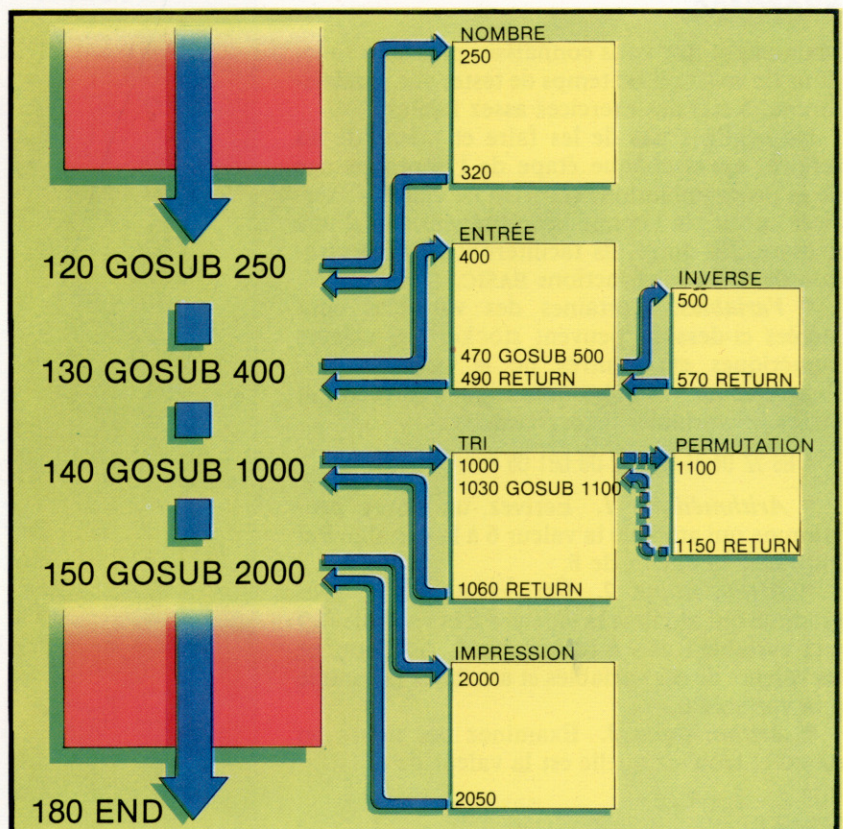
3. INVERSE

Pour inverser l'ordre des noms, nous devons isoler l'« espace » qui sépare le nom du prénom. Nous pouvons utiliser diverses fonctions pour retirer des portions de la chaîne et les attribuer à d'autres chaînes. Les fonctions BASIC effectuent des opérations prédéfinies sur la valeur placée généralement entre parenthèses après le nom de la fonction. La plupart des fonctions sont intégrées mais il est également possible de définir vos propres fonctions. Par exemple, SQR() « renvoie » la racine carrée de la valeur placée entre parenthèses. Ainsi l'instruction : LET A = SQR(9);PRINT A imprimera 3.

INVERSE utilise les fonctions LEN (pour trouver la longueur de la chaîne), INSTR (pour trouver la position de l'espace), LEFT\$ (pour extraire un

Programmes à l'intérieur d'un programme

Cette fois, le programme principal est très court. Tout le travail est fait dans les sous-programmes. Chaque routine du programme a été écrite séparément sous la forme de mini-programmes. Ceux-là sont alors simplement enchaînés par le programme principal. Lorsqu'un programme est exécuté, chaque fois que l'on rencontre une instruction GOSUB, un branchement vers le numéro de ligne spécifié s'effectue et le sous-programme est exécuté. La fin du sous-programme est indiquée par une instruction RETURN. Lorsqu'il atteint cette instruction, le programme revient à l'instruction suivant immédiatement l'instruction GOSUB qui a appelé le sous-programme. Les sous-programmes peuvent être « imbriqués » dans d'autres sous-programmes. Le sous-programme ENTRÉE appelle le sous-programme INVERSE et TRI appelle le sous-programme PERMUTATION. La mise au point des programmes est facilitée par l'emploi de sous-programmes séparés, liés par un programme principal. (Cl. Tony Lodge.)



nombre spécifié de caractères à partir de la gauche de la chaîne) et RIGHT\$ (pour extraire un nombre spécifié de caractères à partir de la droite de la chaîne). Nous verrons les fonctions plus en détail lors de la prochaine partie du cours.

4. TRI

TRI et le sous-programme PERMUTATION sont presque identiques aux sous-programmes utilisés dans la dernière partie du cours.

5. IMPRESSION

Cette étape est très simple :

```
FOR Q = 1 TO N
PRINT A$(Q)
NEXT Q
RETURN
```

Il ne reste plus qu'à écrire le programme principal, aussi simple que cela :

```
REM PROGRAMME PRINCIPAL
GOSUB (NOMBRE)
GOSUB (ENTREE)
GOSUB (TRI)
GOSUB (IMPRESSION)
END
```

Nous avons mis le nom des sous-programmes entre parenthèses. Quelques versions de BASIC appellent des sous-programmes en utilisant leurs noms, mais la plupart utilisent des numéros de ligne. Nous remplacerons donc ces noms par les numéros des lignes appropriées. Certaines instructions REM et PRINT sont aussi ajoutées.

Exercices

Maintenant que vous connaissez plusieurs fonctions de BASIC, il est temps de tester vos connaissances. Voici des exercices assez faciles.

Ne négligez pas de les faire et même de les refaire, après chaque étape de l'apprentissage de la programmation. Car rien ne vaut les exercices pour la bonne compréhension d'une matière. En outre, ils faciliteront la mémorisation des diverses fonctions BASIC.

- **Variables.** Certaines des variables énumérées ci-dessous peuvent stocker des valeurs numériques et d'autres ne le peuvent pas. Encerclez les variables numériques correctes et biffez les variables incorrectes.

A B6 Z2 D\$ 15 X\$ A12 D9 Q81 Q5 6F H\$

- **Arithmétique 1.** Écrivez un court programme qui attribue la valeur 6 à la variable B et imprime la valeur de B.

- **Arithmétique 2.** Écrivez un court programme qui attribue la valeur 5 à la variable A, 7 à la variable B et 9 à la variable C. Additionnez les valeurs de ces variables et attribuez la somme à la variable D.

- **Arithmétique 3.** Examinez ces lignes en BASIC et trouvez quelle est la valeur de C.

```
LET C = 5 + 4 * 3
PRINT C
```

```
10 REM CE PROGRAMME TRIE DES NOMS
20 REM EN ORDRE ALPHABETIQUE
30 PRINT « DECIDEZ D'ABORD COMBIEN »
40 PRINT « DE NOMS VOUS DESIREZ ENTRER »
50 PRINT « PUIS ENTREZ LES NOMS »
60 PRINT « DANS LE FORMAT SUIVANT »
70 PRINT « PRENOM (ESPACE) NOM »
80 REM
90 REM VOICI LE PROGRAMME PRINCIPAL
100 PRINT
110 PRINT
120 GOSUB 250
130 GOSUB 400
140 GOSUB 1000
150 GOSUB 2000
160 REM
170 REM FIN DU PROGRAMME PRINCIPAL
180 END
250 REM SOUS-PROGRAMME TROUVANT LE NOMBRE
260 REM DE NOMS A ENTRER
270 PRINT « COMBIEN DE NOMS »
280 PRINT « DESIREZ-VOUS ENTRER ? »
290 PRINT
300 INPUT N
310 DIM A$(N)
320 RETURN
400 REM SOUS-PROGRAMME D'ENTREE DES NOMS
410 PRINT « ENTREZ LES NOMS SELON CE FORMAT »
420 PRINT « PRENOM (ESPACE) NOM (RETURN) »
430 PRINT « EX : JEAN DUPONT »
440 FOR X = 1 TO N
450 PRINT « ENTREZ LE NOM »
460 INPUT A$(X)
470 GOSUB 500
480 NEXT X
490 RETURN
500 REM SOUS-PROGRAMME INVERSANT LES NOMS
510 LET L = LEN (A$(X))
520 LET S = INSTR(A$(X), " ")
530 LET C$ = LEFT$(A$(X), S - 1)
540 LET F$ = RIGHT$(A$(X), L - S)
550 LET F$ = F$ " "
560 LET A$(X) = F$ + C$
570 RETURN
1000 REM SOUS-PROGRAMME DE TRI
1010 LET S = D
1020 FOR P = 1 TO N - 1
1030 IF A$(P) > A$(P + 1) THEN GOSUB 1100
1040 NEXT P
1050 IF S = 1 THEN GOTO 1000
1060 RETURN
1100 REM SOUS-PROGRAMME DE PERMUTATION
1110 LET T$ = A$(P)
1120 LET A$(P) = A$(P + 1)
1130 LET A$(P + 1) = T$
1140 LET S = 1
1150 RETURN
2000 REM SOUS-PROGRAMME D'IMPRESSION
2010 PRINT
2020 FOR Q = 1 TO N
2030 PRINT A$(Q)
2040 NEXT Q
2050 RETURN
```

• **Arithmétique 4.** Quels nombres seraient imprimés par ce programme ?

```
LET A = 3
LET B = 2
LET C = 9
LET D = 4
LET E = (A + B) * (C - D)
PRINT E
LET E = 5
LET E = E * E
PRINT E
```

• **Comparaisons 1.** Que doit valoir X pour que le message PRINT soit imprimé ?

```
70 LET A = 5
80 LET B = X
90 LET R = B - A
100 IF R = 0 THEN GOTO 120
110 GOTO 10
120 PRINT « FELICITATIONS! VOUS GAGNEZ »
999 END
```

• **Comparaisons 2.** Quelle est la valeur minimale de X qui fera passer le programme à la ligne 300 ?

```
250 IF X > 6 * 100 THEN GOTO 300
```

• **Comparaisons 3.** Quelle est la valeur minimale de Z qui fera passer le programme au message « félicitations ? »

```
340 IF Z < 10000 THEN GOTO 500
350 IF Z >= 10000 THEN GOTO 520
:
:
500 PRINT « TROP BAS, ESSAYEZ ENCORE »
510 GOTO 600
520 PRINT « FELICITATIONS »
530 GOTO 700
```

• **Impression 1.** Supposons que T = 50. Écrivez une instruction PRINT qui imprimera : LA VALEUR DE T EST 50. Conseil : mettre le « message » entre guillemets, utiliser un point-virgule et un nom de variable.

• **Impression 2.** Examinez le court programme suivant et complétez l'instruction PRINT afin que le programme imprime le message :

```
DESOLE, VOTRE POINTAGE 175 EST TROP BAS
```

Écrivez cette ligne de façon que la valeur du pointage puisse varier.

```
600 REM LA VARIABLE S EST LE POINTAGE
620 IF S <= 500 THEN GOTO 640
630 GOTO 700
640 PRINT « DESOLE »
```

• **Impression 3.** Quel message sera imprimé lors de l'exécution de ce programme ?

```
200 LET A$ = « LE COURS DE MICRO-INFORMATIQUE ? »
210 LET B$ = « AIMEZ-VOUS »;
220 PRINT B$
230 PRINT A$
```

• **Entrée 1.** L'instruction INPUT est une façon d'affecter une valeur à une variable. Dans le programme suivant, quelle touche doit être pressée pour afficher 12 comme réponse ?

```
60 INPUT N
70 LET N = N * 2
80 PRINT N
```

• **Entrée 2.** Qu'est-ce qui sera imprimé ici ?

```
100 PRINT « TAPEZ VOTRE NOM SVP »
110 INPUT N$
120 PRINT « BONJOUR »; N$; « ICI VOTRE ORDINATEUR »
```

Variantes de basic

Ce programme ne fonctionnera pas sur le Lynx qui n'accepte pas les tableaux de chaînes. Il ne fonctionnera pas non plus sur l'Atari 400 ou 800, puisque leur manipulation de chaîne est différente.

L'utilisation de cette instruction n'est pas standard sur le Spectrum.

DIM

GOTO

La plupart des ordinateurs vous permettent d'omettre le mot GOTO; la ligne 1050 peut donc être écrite :

```
1050 IF S = 1 THEN 1000
```

INSTR

Si cette fonction n'est pas disponible sur votre ordinateur, effacez la ligne 520 et remplacez-la par ces cinq lignes :

```
515 FOR P = 1 TO L
520 CH$ = MID$(A$(X), P, 1)
522 LET S = 0
523 IF CH$ = " " THEN LET S = P :
LET P = L
525 NEXT P
```

INSTR

Aucune de ces commandes n'est disponible sur le Spectrum, mais vous pouvez créer vos propres versions avec l'instruction DEF FN.

```
320 DEF FN M$(X$,N) = X$(N)
330 DEF FN L$(X$,N) = X$(TO N)
340 DEF FN R$(X$,N) = X$(N TO)
350 RETURN
```

LEFT\$

puis effacez les lignes 510 à 560 et remplacez-les par les lignes suivantes :

```
510 LET D$ = A$(X)
520 LET L = LEN(D$)
530 LET S = 0
540 FOR P = 1 TO L
550 IF FN M$(D$,P) = " " THEN
LET S = P
560 IF S < > 0 THEN LET P = L
570 NEXT P
580 LET C$ = FN L$(D$, S - 1)
590 LET F$ = FN R$(D$, L - S)
600 LET G$ = F$ + " " + C$
610 LET A$(X) = G$
620 RETURN
```

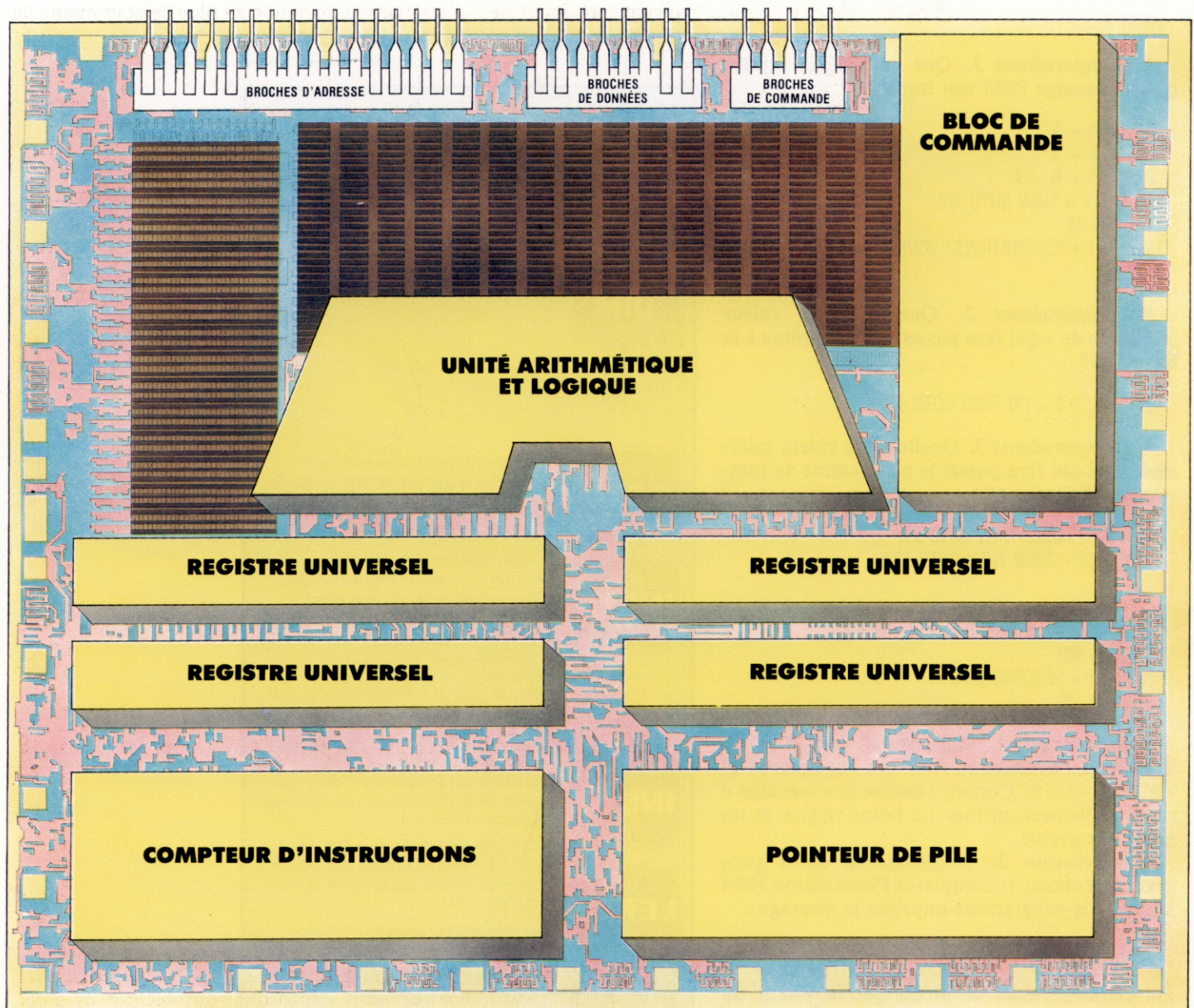
RIGHT\$

END

Cette instruction n'est pas disponible sur le Spectrum, l'Oric-1 ou le Dragon 32, remplacez-la par STOP.

Le centre nerveux

Toutes les interventions de l'ordinateur sont gérées par l'unité centrale de traitement.



Le fonctionnement de l'ordinateur est commandé par l'UC (unité centrale). Le terme « unité centrale » est adéquat puisque ce composant représente le cœur de l'ordinateur. Un ordinateur très simple pourrait n'être composé que d'une UC, d'une mémoire et de circuits d'E/S (entrée/sortie).

L'ordinateur a besoin des circuits d'E/S pour communiquer avec l'extérieur. Dans une application très simple, un ordinateur de machine à laver nécessite une entrée/sortie pour mettre en marche les moteurs et les éléments de chauf-

fage. La mémoire, vive ou morte, sert à stocker des instructions et des données destinées à l'UC. Les données traitées par l'UC peuvent être des nombres et des codes binaires représentant des caractères (lettres, chiffres et symboles comme \$ et *).

Si la mémoire renferme à la fois des données et des instructions pour l'UC, comment fait-elle pour discerner les données à traiter des instructions à exécuter? Pour répondre à cette question, nous devons examiner l'intérieur du micro-ordinateur.

Les UC des ordinateurs à 8 bits (la plupart des ordinateurs domestiques) se présentent généralement dans une puce munie de quarante broches, vingt de chaque côté. Chacune de ces broches (à l'exception des broches d'alimentation 0 et + 5 V) transmet des signaux entre l'UC et d'autres dispositifs comme la mémoire ou les circuits d'E/S.

Les UC ont généralement seize broches d'« adresse » connectées au « bus d'adresse ». Chacune de ces broches transmet un signal de sortie représentant soit un 1 soit un 0, ce qui donne 65 536 combinaisons de 1 et de 0. Ces combinaisons servent à spécifier des adresses dans la mémoire.

Il y a aussi huit broches de « données » connectées au « bus de données ». Les broches de données transmettent des données entre l'UC et la mémoire ou les circuits d'E/S.

D'autres broches transmettent des signaux de commande. Certains sortent de l'UC, d'autres y entrent. Nous verrons bientôt comment sont utilisés les signaux de commande.

Dans l'UC, il y a quelques cellules de mémoire à un ou deux octets que l'on nomme registres et dont certaines sont réservées à des fonctions spécifiques. Les autres servent à stocker temporairement les informations et sont appelées registres universels.

Il y a deux autres « blocs » importants dans l'UC, l'ALU et le « bloc de commande ».

Le terme « ALU » signifie unité arithmétique et logique. Cette partie de l'UC se charge des opérations arithmétiques et logiques comme l'addition, les opérations ET et OU et le déplacement des bits dans un octet.

Le bloc de commande est un circuit spécial servant à faire exécuter par l'UC les instructions venant de la mémoire. Prenons un exemple précis concernant les codes d'instructions du populaire Z80. Si l'instruction codée en provenance de la mémoire est 11000110, l'UC additionnera le contenu du prochain octet en mémoire au contenu de l'un des registres de l'UC. Si nous

désirons stocker le résultat de cette addition dans une adresse spécifique de la mémoire, la prochaine instruction reçue par l'UC devra être 00110010, suivie de deux octets spécifiant l'adresse de la mémoire où le résultat doit être stocké.

Supposons que le résultat de l'addition est 37 (en notation décimale), et que les deux octets suivant l'instruction de stockage spécifient l'adresse 33126 (toujours en décimal). Le code d'instruction demande au bloc de commande de régler les broches d'adresse à l'équivalent binaire de 33126 (1000000101100110). Les broches de commande envoient des signaux demandant à la mémoire de recevoir et de stocker des données. Les broches de données sont alors réglées à l'équivalent binaire de 37 (00100101). Ces données sont ensuite transmises à la mémoire par le bus de données et sont stockées à l'adresse spécifiée par le bus d'adresse. Si, plus tard, l'UC a besoin de ces données pour poursuivre le traitement (lors d'un affichage par exemple), une autre instruction est envoyée à l'UC. Le bloc de commande interprète cette instruction comme étant « lire l'octet de l'adresse 33126, le stocker temporairement dans l'un des registres internes ».

Le nombre de registres à l'intérieur de l'UC est variable. Il peut s'agir de registres à 8 bits ou à 16 bits. Les registres spécialisés portent des noms comme « pointeur de pile », « compteur d'instruction » ou « accumulateur ». On désigne généralement les registres universels comme ceci : « registre X », « registre Y », « registre C » et ainsi de suite.

Un des registres à 16 bits les plus importants est le registre « compteur d'instructions », qui contient toujours l'adresse (binaire) de la prochaine instruction à exécuter en mémoire. Lorsqu'il est temps de lire l'instruction suivante, le contenu du compteur d'instructions est mis sur le bus d'adresse et l'octet de cette adresse est transmis à l'UC (via le bus de données).

Le registre à 8 bits le plus important est l'« accumulateur ». Ce registre stocke généralement (temporairement) le résultat des opérations effectuées par l'ALU, les octets provenant de la mémoire ou de l'E/S, ou l'adresse où des octets sont stockés momentanément, juste avant d'être envoyés à la mémoire ou à l'E/S.

Cette introduction aux UC a été plutôt sommaire; nous approfondirons bientôt ce sujet. Nous avons vu ici que des instructions spéciales lues dans la mémoire font effectuer certaines opérations à l'UC et règlent les broches d'adresse afin d'accéder à des adresses de mémoire spécifiques. Les données sont extraites à ces adresses, ou y sont envoyées par le bus de données. Des signaux du bus de commande indiquent aussi à la mémoire ou à l'E/S si les données doivent être « lues » ou « écrites ».

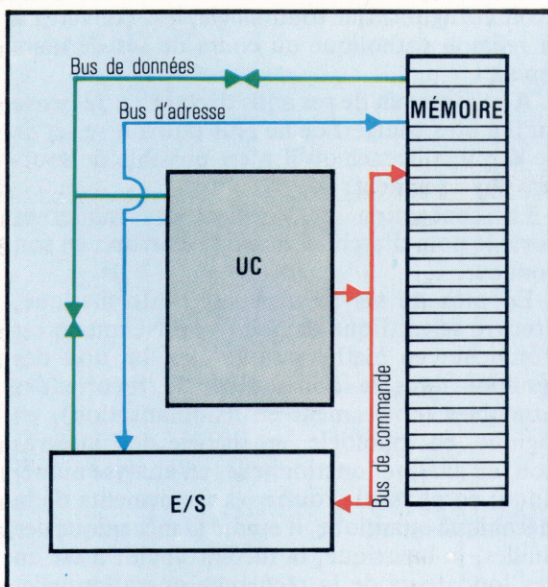
Bientôt, nous vous apporterons d'autres éléments de réponse aux questions que vous vous posez au sujet du fonctionnement de l'UC et de son rôle dans le traitement de l'information.

Sous contrôle

Ceci illustre une UC comportant des « registres » de mémoire, une unité arithmétique et logique composée de centaines de portes logiques (qui effectuent des opérations du type addition, ET ou OU) et un bloc de commande. Le bloc de commande accepte l'instruction codée (en binaire), l'interprète et gère l'intervention des autres composantes de l'UC. Par exemple, si une instruction implique que le contenu de l'accumulateur doit être stocké à une adresse de mémoire particulière, le bloc de commande définira cette adresse sur les broches d'adresse, enverra des signaux de commande demandant à la mémoire de stocker des données, et mettra le contenu de l'accumulateur sur le bus de données afin de le transmettre vers la mémoire. (Cl. Kevin Jones.)

Exécution des ordres

Un ordinateur très simple peut ne comporter qu'une UC, une mémoire et un circuit d'E/S. La mémoire stocke les instructions qui commandent les diverses interventions de l'UC. Elle stocke également des données devant être traitées selon les instructions. Le circuit d'E/S permet à l'UC de communiquer avec le monde extérieur. Si l'ordinateur commande une machine à laver, le circuit d'E/S entrera des signaux en provenance des boutons du panneau de commande et enverra des signaux de sollicitation vers les moteurs et les éléments de chauffage. Les codes d'instruction de l'UC sont formulés en binaire. Chaque modèle différent d'UC possède son propre jeu de codes d'instruction (code machine).



John von Neumann

Ce brillant mathématicien d'origine austro-hongroise a participé à la conception de l'ordinateur moderne.



Un couple bien assorti
John von Neumann en compagnie de sa seconde épouse, Klara, elle-même excellent programmeur des premiers ordinateurs. (Cl. A.P.)

L'esprit de compétition des Hongrois est proverbial, et John von Neumann ne fait certes pas exception. Son ambition et son intelligence extraordinaire l'ont placé aux plus hauts postes scientifiques américains.

Neumann était issu d'une riche famille juive de l'Empire austro-hongrois. Ses talents mathématiques se manifestèrent rapidement et, à l'âge de vingt-cinq ans, il obtenait un doctorat et discutait sur un pied d'égalité de problèmes mathématiques avec des hommes tels qu'Einstein ou David Hilbert.

À la chute de l'Empire austro-hongrois, il adopta la particule « von » et s'installa en Allemagne. Simultanément, il développa de nombreux contacts aux États-Unis, passant l'hiver à l'université de Princeton (New Jersey) et l'été en Europe où il administrait les propriétés de son père.

Lorsque la Seconde Guerre mondiale éclata, il se réfugia aux États-Unis.

Von Neumann participa à l'établissement de la théorie des ensembles en poursuivant les travaux de Bertrand Russell. Il était fasciné par la mécanique quantique et la théorie des jeux. Il inventa la méthode Monte-Carlo, qui utilise des nombres aléatoires pour résoudre des équations mathématiques.

Lorsque les États-Unis s'engagèrent dans la guerre, il fut immédiatement affecté au projet Manhattan et participa activement aux travaux théoriques qui ont permis la production de la bombe atomique.

Il était toujours impliqué dans le projet Manhattan lorsqu'il apprit qu'on tentait de cons-

truire un ordinateur électronique. Il offrit immédiatement son concours au projet ENIAC, qui était sous la responsabilité d'ingénieurs électroniciens. Comme il était le premier mathématicien à vraiment prendre part au projet, il rédigea un rapport qui devint l'esquisse de l'ordinateur moderne.

Après la guerre, il s'impliqua de plus en plus dans la défense des États-Unis. Mais il poursuivit ses recherches mathématiques et conçut le premier ordinateur de l'université de Princeton, nommé JOHNIAC.

Vers la cinquantaine, il fut atteint d'un cancer et cloué dans un fauteuil roulant. Après avoir été agnostique toute sa vie, il se convertit à la religion catholique au cours de ses derniers mois.

À sa mort, un de ses amis déclara : « Je pense qu'il a plus souffert de ne plus pouvoir se servir de son intelligence qu'il n'est possible de souffrir physiquement. »

La conception des ordinateurs modernes porte le nom d'architecture de Neumann en son honneur.

En plus de ses travaux sur l'informatique, l'œuvre scientifique de John von Neumann est très riche; en mathématiques, on lui doit des résultats dans le domaine de la théorie des ensembles (notamment en axiomatisation), en logique, en topologie, en théorie de l'intégration, en analyse fonctionnelle, en analyse numérique; en physique, outre les fondements de la mécanique quantique, il étudie la mécanique des fluides, la balistique, la météorologie; il est un des fondateurs de la recherche opérationnelle.

1903

Janos Louis Neumann naît à Budapest le 28 décembre.

1933

Il devient professeur à l'université de Princeton, New Jersey, le paradis d'Einstein et de nombreux autres scientifiques européens émigrés.

1943

Il participe au projet Manhattan à Los Alamos, Nouveau-Mexique.

1944

Il entend parler du projet secret ENIAC.

1945

Il assiste aux premiers essais nucléaires.

1947

Il écrit un rapport pour ENIAC et conçoit un nouvel ordinateur nommé EDVAC (*Electronic Discrete Variable Computer*) devant être construit selon les propositions contenues dans le rapport.

1951

Il conçoit lui-même JOHNIAC, qui est utilisé par l'université de Princeton.

1951-1953

Président de la Société américaine de mathématiques. Il s'intéresse à la cybernétique.

1955

Nommé commissaire à l'énergie atomique des États-Unis par le président Eisenhower.

1957

Il meurt d'un cancer le 8 février. Un volume complet du *Bulletin of the American Mathematical Society* est consacré à sa vie et à ses travaux.

Demandez le programme

Chaque semaine,
ABC informatique
vous présentera dans ses
pages 3 et 4 de couverture
des programmes de jeux, de calcul
ou de gestion personnelle à réaliser
sur votre micro-ordinateur.
Au terme de cette encyclopédie,
vous disposerez ainsi d'une
série d'exercices de toute nature,
utiles et distrayants, dont la
difficulté ira croissant.
Si vous avez des questions à nous
poser à leur propos, n'hésitez pas à nous
écrire. Notre rédaction vous aidera
à résoudre vos problèmes.

Le BASIC (Beginner's All purpose Symbolic Instruction Code) est un langage mis au point par des universitaires américains de Dartmouth; dérivé du FORTRAN, il connaît actuellement un grand développement en raison du formidable essor des micro-ordinateurs professionnels ou personnels. C'est désormais le langage le plus employé dans l'informatique. On en trouve l'utilisation aussi bien pour la gestion, l'acquisition des données, les calculs scientifiques que pour les applications graphiques ou le contrôle de processus.

Or la programmation s'apprend essentiellement par la pratique.
Dès cette semaine, nous vous proposons une collection complète d'exercices concrets.
Chaque exercice comporte toutes les étapes du processus :

- énoncé du problème;
- analyse;
- organigramme avec commentaires;
- listage de programme;
- exemples d'exécution.

Nous espérons qu'après avoir effectué un nombre suffisant de ces exercices et en avoir eu l'explication vous pourrez dire : « Je connais le BASIC. »

Cette méthode permet au lecteur de mieux structurer sa réflexion, le guidant, ainsi, dans l'élaboration d'un programme personnel; elle lui enseigne également comment décomposer un problème en « sous-problèmes » plus simples, pour en sérier les difficultés et traiter ces dernières séparément, grâce à autant de sous-programmes.

Ces sous-programmes peuvent ensuite servir de base à l'élaboration de programmes plus ambitieux et plus pointus.

Pour chaque programme, toute nouvelle instruction utilisée est accompagnée de sa définition détaillée.

Pour la majorité des programmes, le BASIC retenu sera le BASIC de Microsoft (ou MBASIC). Pour quelques autres, plusieurs versions pourront être proposées afin de montrer comment pallier les « manques » de certains BASIC.

PROGRAMME N°1

VOTRE PREMIER PROGRAMME EN BASIC

Ce programme démontre que tout un chacun peut programmer en BASIC et que, du moins pour certaines catégories de problèmes, point n'est besoin d'être un spécialiste. Vous allez apprendre les premières instructions du BASIC et comment progressivement améliorer ce programme.

THÈME

Vous désirez calculer la somme de deux nombres.

Appelons A un premier nombre,

— B un second nombre,

— S la somme de A et B.

Pour obtenir S, il suffit de faire effectuer la somme $A + B$.

On va donc concevoir un programme qui effectuera les opérations suivantes :

Lire A 10 INPUT A

Lire B 20 INPUT B

Calculer $S = A + B$ 30 $S = A + B$
Imprimer S 40 PRINT S
50 END

Ce court programme appelle quelques brèves remarques.

- Chaque ligne reçoit un numéro de ligne : en effet, le micro-ordinateur exécute les instructions du programme dans l'ordre croissant des numéros de ligne.

- Chaque ligne comporte une instruction :
— l'instruction INPUT (littéralement, « mettre dans ») correspond à l'entrée d'un traitement (ici, S);

- l'instruction de calcul $S = A + B$ s'écrit sur une seule ligne, + étant le symbole de l'addition. Les autres symboles opératoires sont : — (moins) pour la soustraction, * (astérisque) pour la multiplication et / (barre de fraction, ou « slash ») pour la division.

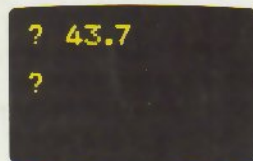
- Le programme se termine par END.

EXÉCUTION DU PROGRAMME

Le programme étant tapé dans son intégralité, le faire exécuter par l'instruction RUN. On voit alors apparaître sur l'écran :



La machine réclame par ? la valeur de A. L'utilisateur tape un chiffre quelconque, 43,7, par exemple; apparaît alors :



La machine réclame alors par le second ? l'entrée d'un nombre B. L'utilisateur tape un nombre quelconque, 56,3, par exemple.

L'entrée de chaque nombre A et B doit être suivie de la touche RETURN. Après la saisie de B apparaît sur l'écran :



Lorsque l'ordinateur doit exécuter l'instruction INPUT, il édite sur l'écran un ? afin d'avertir l'utilisateur que celui-ci peut envoyer les données. Ici, le résultat est correct du point de vue mathématique mais peu attrayant quant à la présentation.

La semaine prochaine, nous verrons comment rendre plus élégante l'exécution de ce petit programme; nous verrons aussi ce que représente un organigramme et comment l'écrire.

En attendant, vous pouvez continuer à vous familiariser avec les instructions INPUT et PRINT en calculant, par exemple, la somme de trois nombres ou toute autre combinaison d'opérations de votre choix.

**A LA SEMAINE
PROCHAINE!**